1977

The CQDRNG8 - a quadratic, isoparametric, axisymmetric finite element for the NASTRAN computer program

John Gray

THE CQDRNG8 - A QUADRATIC, ISOPARAMETRIC, AXISYMMETRIC FINITE ELEMENT FOR THE NASTRAN COMPUTER PROGRAM

by

John F. Gray

A Thesis Submitted

in

Partial Fulfillment

of the

Requirements for the Degree of

MASTER OF SCIENCE

in

Mechanical Engineering

Approved by:

Prof. Wayne Walters

Prof. B. Karlekar

Prof. Neville Rieger

Prof. Richard B. Hetnarski

DEPARTMENT OF MECHANICAL ENGINEERING

ROCHESTER INSTITUTE OF TECHNOLOGY

ROCHESTER, NEW YORK

May 1977

ACKNOWLEDGEMENTS

The success of this undertaking is largely due to the cooperation of many persons. I would like to express my deepest gratitude to the following:

To my wife Phyllis and my son Kevin for their patience and understanding throughout this time period.

To Dr. V. Genberg and Mr. P. Mueller, Eastman Kodak Company, for their suggestions and technical advice.

To S. Wall, J. Joseph, and R. Harder, The MacNeal-Schwendler Corporation, for their gracious assistance in providing needed insight into the MSC/NASTRAN program.

To Professors W. Walter, my thesis adviser, and N. Rieger, Rochester Institute of Technology, Mechanical Engineering Department, for their guidance and suggestions.

To Eastman Kodak Company for use of the computer facilities and access to the MSC/NASTRAN program.

ABSTRACT

The development of an axisymmetric ring finite element is presented and FORTRAN subroutines for implementing the capability into the MSC/NASTRAN finite element program are given. The element is an eight-noded isoparametric quadrilateral of quadratic order. The following matrices and capabilities are developed:

1. stiffness matrix for homogeneous isotropic materials,
2. thermal conductance matrix for homogeneous isotropic materials,
3. calculation of equivalent nodal forces due to temperature loads,
4. calculation of stresses, and
5. plotting of undeformed and deformed structures.

Several classical thermal and structural problems are solved to demonstrate these capabilities. In all cases, the element results compare well with theory. Comparisons are made to existing MSC/NASTRAN axisymmetric finite elements. The new element shows increased accuracy compared to the existing elements. Convergence of the element is shown.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

r,θ,z	Cylindrical coordinates (radial, circumferential, axial)
$\emptyset$	Strain energy
$\sigma_r,\sigma_\theta,\sigma_z$ τ_{rz}	Normal stresses in r,θ,z directions and shearing stress in rz plane
$\varepsilon_r,\varepsilon_\theta,\varepsilon_z$ γ_{rz}	Normal strains in r,θ,z directions and shearing strain in rz plane
E	Young's modulus
ν	Poisson's ratio
VOL	Volume
u,v	Radial and axial displacement components
N_i	Shape function associated with node i
α,β	Local element coordinates
U	Thermal potential function
$\tilde{q}$	Heat flux density
k	Thermal conductivity
x_i	Generalized coordinate
$\{\ \}$	Vector
$[\ ]$	Matrix
$[\ ]^{-1}$	Inverse of a matrix
$\{\ \}^T,[\ ]^T$	Transpose of vector or matrix
$\lvert\ \rvert$	Determinant of a matrix
$[K]$	Stiffness or conductance matrix
$\{u\}$	Nodal vector of unknowns (displacements or temperatures)
$\{P\}$	Nodal vector of applied loads
$\{\sigma\}$	Element stress vector
$\{\varepsilon\}$	Element strain vector

$[D]$	Constitutive matrix relating stress and strain
$[B]$	Matrix relating strains to displacements
$[J]$	Jacobian matrix
$[T]$	Transformation matrix
W_{ij}	Gaussian integration weighting factor

I. INTRODUCTION

NASTRAN is a large general purpose finite element program with a substantial element library. One of the most noticeable deficiencies is the absence of a higher order quadrilateral axisymmetric ring element. The objective of this thesis is to fill this void. A suitable element with limited capabilities was developed by Janucik.[1] This effort is an extension of that development. Additional capabilities are developed and this element is implemented in the NASTRAN program as a DUMMY element.

This element, named the CQDRNG8, has many capabilities. Elastic stiffness is calculated for homogeneous isotropic materials. Thermoelastic loading is supported. Plots for both undeformed and deformed structures can be obtained. Element stresses are ouput at the centroid and the four corner grids of each element. Stress invariants calculated include the principal stresses and directions, the maximum shear, and the octahedral shear. A thermal conductance matrix is calculated for homogeneous isotropic materials thus providing a steady state heat transfer capability. The CQDRNG8 is an eight noded isoparametric quadrilateral element which allows quadratic variation of displacement and temperature within the element. A number of user options exist with respect to both NASTRAN and this element.

The CQDRNG8 element's capabilities are illustrated by solving several classical thermal and structural problems. In all cases, very good correlation is observed between theoretical and finite element solutions. Convergence of the solution for increasingly detailed finite element models is demonstrated. Comparisons with existing NASTRAN axisymmetric

elements is made.

Data cards for the new element and subroutines for implementing the CQDRNG8 into NASTRAN are given in the appendices.

II. LITERATURE REVIEW

In the last decade, many texts and articles on finite element theory have been published. Perhaps the most widely referenced texts are those by Zienkiewicz.[2,3] In his texts, Zienkiewicz develops the general theory for finite element analysis. His texts also include individual sections on specific finite element formulations. A comprehensive treatment of the use of isoparametric shape functions is included. Finally Zienkiewicz devotes a discussion to solving general field problems using finite element methods. Particular references are made to the heat transfer problem.

The text by Przemieniecki[4] addresses itself to the solution of structural problems using matrix techniques. Some simple finite element formulations are developed. This text includes a thorough treatment of matrix substructuring -- the reduction of matrix size by partitioning operations.

Segerlind[5] presents a finite element text which is not structures oriented. Finite element formulations for heat transfer, fluid mechanics, and elasticity problems are developed. Axisymmetric field problems (both steady state and transient) are discussed. Computer implementation of the various finite element formulations is discussed. A number of instructional computer programs are listed.

Desai[6] develops the finite element matrices for the structural problem. Techniques for nonlinear analysis are discussed. He presents a number of useful examples.

Bathe's[7] finite element developments are rather brief but he presents a comprehensive treatment of matrix solution techniques. Of particular interest are elimination techniques for equation solution, eigenvalue extraction methods, and transient integration routines.

In his paper, Ergatoudis[8] describes a family of isoparametric quadrilateral finite elements. He presents isoparametric shape functions for linear, quadratic, and cubic elements. Special variations include additional internal degrees of freedom and a mixed order finite element which can have different numbers of grids along each edge. Baldwin[9] gives the computer routines for a quadratic order isoparametric thin plate element and demonstrates the accuracy of the element for the solution of bending problems.

Detailed derivations of finite element matrices for heat transfer problems are given by Lee[10] and MacNeal.[11] Both authors develop the conductance and capacitance matrices for the finite elements in NASTRAN. MacNeal also develops the structural matrices for the NASTRAN elements. Lee presents several examples illustrating NASTRAN's heat transfer capabilities.

Doherty's[12] paper develops several higher order axisymmetric finite elements. Quadrilateral elements are formed by combining various numbers of triangular elements then eliminating interior grid points. In an effort to avoid midside grids, mixed order triangular elements are combined such that a quadratic displacement function exists interior to the element but only a linear variation occurs along exterior edges. The developments include the effects of orthotropic and temperature dependent materials.

Both Bruch[13] and Kohler[14] develop the finite element matrices for the solution of transient two dimensional heat conduction problems. Bruch uses a rectangular element with linear temperature variation. Kohler uses general quadrilateral elements having linear, quadratic, and cubic temperature variations. In his paper, Zienkiewicz[15] presents higher order isoparametric finite element formulations for the solution of two and three dimensional transient field problems. He develops a transient solution formula and solves a heat conduction problem.

The basis for this work is a thesis submitted by Janucik[1]. In his paper, Janucik develops the stiffness matrix for the eight noded isoparametric quadrilateral ring element. He presents a stand alone computer program for utilizing this capability. He describes several limitations relating to this program including numerical precision, core, and problem size constraints. The current work is designed to enhance the capabilities of this element and to eliminate the problems by adding the element to the NASTRAN finite element program.

The NASTRAN texts by Wall[16,17] are the basis for designing and adding the software to realize this new element capability. A thorough description of the general NASTRAN techniques used in formulating and solving the desired equations is presented. The texts outline the specific capabilities needed. Detailed formats of the tables and matrices needed by the programmer are presented along with instructions for utilizing available NASTRAN capabilities. The extensive NASTRAN overlay structure is explained. Techniques for implementing the new element capability within this framework are given.

The manuals by McCormick[18,19] provide an introduction to the capabilities and use of the NASTRAN computer program. Detailed user information is found in the first manual. Additional NASTRAN features and special modeling considerations are described by Joseph.[20] NASTRAN numerical techniques and element formulations are described by MacNeal.[11]

III. AXISYMMETRIC STRUCTURES

Axisymmetric structures under axisymmetric loading are frequently encountered in engineering problems. These three dimensional structures can efficiently be analysed as two dimensional ones by properly treating the components of displacement and strain. The symmetry of both structure and loading allow the behaviour to be completely described by considering only a cross section of the structure. This cross section must pass through the axis of axisymmetry and be normal to the circumferential direction. The symmetry requires that the circumferential displacement be zero. In the development of this element, the restriction is imposed that the axis of axisymmetry be the z axis and that the cross section considered lie in the x-z plane. In this paper, the cylindrical coordinates (r,θ,z) will be employed. The x coordinate then is the radial coordinate r and the circumferential coordinate becomes θ. The z coordinate remains unchanged. The displacement components of interest are the r and z translations. In the context of this geometry, four components of stress are defined -- the three normal stresses (σ_r, σ_θ, σ_z) and the shear stress in the plane (τ_{rz}). Due to the symmetry, the shear stress in the r-z plane is the only non-zero shear stress.

In heat transfer, symmetric thermal loading requires that the heat flow in the circumferential direction be zero. The two non-zero components of heat flow are in the r and z directions.

In finite element analysis, this cross section is modeled using axisymmetric elements. Axisymmetric finite elements are really rings of revolution. Each element describes a cross sectional area which is re-

volved through 360° to form a solid ring. The complete volume is represented by modeling the right half of the cross section (r≥0). Axisymmetric loads applied to grid points represent the total force (or heat flow) applied to the circular arc formed by revolving the grid through 360°.

IV. ELEMENT DERIVATIONS

A. STIFFNESS MATRIX

In statics, the equation being solved is

$$[\,K\,]\,\{\,u\,\} = \{\,P\,\} \qquad (1)$$

where $[\,K\,]$ is the structural stiffness matrix,
$\{\,u\,\}$ is the vector of nodal displacements, and
$\{\,P\,\}$ is the vector of applied forces.

The stiffness matrix can be derived from the structural potential function -- the strain energy. In matrix notation, the strain energy in an element is

$$\emptyset = \tfrac{1}{2} \int \{\,\sigma\,\}^{T} \{\,\varepsilon\,\}\ \underline{d}VOL \qquad (2)$$

where $\{\,\sigma\,\}$ is a vector of internal stresses,
$\{\,\varepsilon\,\}$ is a vector of internal strains, and
the integration is performed over the volume.

Castigliano's Theorem states that the derivative of the strain energy with respect to the displacements is equal to the applied forces. Writing Castigliano's Theorem and substituting from equation (1)

$$\frac{\partial \emptyset}{\partial \{\,u\,\}} = \{\,P\,\} = [\,K\,]\,\{u\} \qquad (3)$$

Before the derivative can be evaluated, the strain energy must be written in terms of the displacements.

First the constitutive equation relating $\{\,\sigma\,\}$ and $\{\,\varepsilon\,\}$ can be written

$$\{\,\sigma\,\} = [D\,]\ \{\,\varepsilon\,\} \qquad (4)$$

Four components of stress and strain are defined for axisymmetric

structures:

$$\{ \sigma \} = \begin{Bmatrix} \sigma_r \\ \sigma_\theta \\ \sigma_z \\ \tau_{rz} \end{Bmatrix} \quad ; \qquad \{ \varepsilon \} = \begin{Bmatrix} \varepsilon_r \\ \varepsilon_\theta \\ \varepsilon_z \\ \gamma_{rz} \end{Bmatrix}$$

The constitutive matrix $[D]$ is given in Timoshenko[21] as

$$\frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \qquad (5)$$

Substituting into the expression for the strain energy yields

$$\phi = \tfrac{1}{2} \int \{ \varepsilon \}^T [D] \{ \varepsilon \} \, dVOL \qquad (6)$$

Next an expression relating strains to displacements is written. The four components of strain are defined as

$$\begin{Bmatrix} \varepsilon_r \\ \varepsilon_\theta \\ \varepsilon_z \\ \gamma_{rz} \end{Bmatrix} = \begin{Bmatrix} \frac{\partial u}{\partial r} \\ \frac{u}{r} \\ \frac{\partial v}{\partial z} \\ \frac{\partial u}{\partial z} + \frac{\partial v}{\partial r} \end{Bmatrix} \qquad (7)$$

where u and v are displacement components.

Continuous displacement functions are written using isoparametric shape functions

$$u = \sum_i N_i \; u_i$$
$$v = \sum_i N_i \; v_i \qquad (8)$$

where u_i and v_i are displacement components at the i^{th} grid point, and N_i is the shape function associated with the i^{th} grid point.

The element coordinate system and the shape functions for an eight noded isoparametric quadrilateral element as defined by Zienkiewicz[2] are shown in Figure 1. Arbitrarily shaped elements are mapped into the 2 by 2 square shown by the same shape functions:

$$r = \sum_i N_i(\alpha,\beta) \; r_i$$
$$z = \sum_i N_i(\alpha,\beta) \; z_i \qquad (9)$$

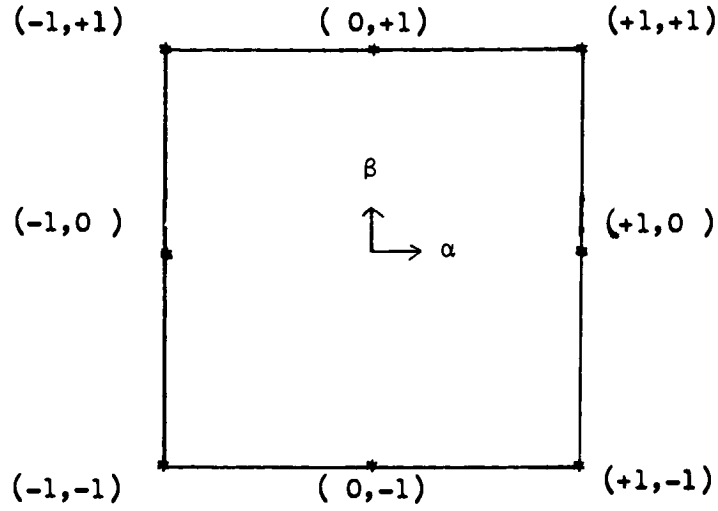
where r_i and z_i are the r, z coordinates at node i.

Since the shape functions vary quadratically in α and β, the displacements also vary quadratically within the element. The mapping of coordinates allows curved geometries to be modeled. This concept gives rise to the name isoparametric -- literally "same parameters". The same parameters are used to describe both the geometry and the displacements. One of the properties of the shape functions is that at grid i, the i^{th} shape function has a value of 1.0 and all other shape functions are identically zero. As described by Zienkiewicz, isoparametric shape functions inherently satisfy the conditions necessary for convergence. These conditions are:

1) the element is strain free for rigid body motion, and
2) a constant strain condition can be represented.

FIGURE 1. ELEMENT COORDINATE SYSTEM AND SHAPE FUNCTIONS

Element coordinate system



Element shape functions

Corner nodes (odd indices)

$$N_i = \tfrac{1}{4}(1+\alpha_o)(1+\beta_o)(\alpha_o+\beta_o-1)$$

$$\frac{\partial N_i}{\partial \alpha} = \tfrac{1}{4}\alpha_i(1+\beta_o)(2\alpha_o+\beta_o)$$

$$\frac{\partial N_i}{\partial \beta} = \tfrac{1}{4}\beta_i(1+\alpha_o)(\alpha_o+2\beta_o)$$

Midside nodes (even indices)

$$N_i = \tfrac{1}{2}(1-\alpha^2)(1+\beta_o)\beta_i^2 + \tfrac{1}{2}(1-\beta^2)(1+\alpha_o)\alpha_i^2$$

$$\frac{\partial N_i}{\partial \alpha} = -\alpha(1+\beta_o)\beta_i^2 + \tfrac{1}{2}(1-\beta^2)\alpha_i$$

$$\frac{\partial N_i}{\partial \beta} = -\beta(1+\alpha_o)\alpha_i^2 + \tfrac{1}{2}(1-\alpha^2)\beta_i$$

where $\alpha_o = \alpha\alpha_i$ and $\beta_o = \beta\beta_i$

α_i, β_i are the values of coordinates α, β at node i

Substituting the displacement relations into equation 7 yields the strain displacement relations

$$\{ \epsilon \} = [B] \{ u \} \qquad (10)$$

where the matrix $[B]$ is defined as

$$[B] = \begin{bmatrix} \frac{\partial N_1}{\partial r} & 0 & \frac{\partial N_2}{\partial r} & 0 & \frac{\partial N_3}{\partial r} & 0 & \frac{\partial N_4}{\partial r} & 0 & \frac{\partial N_5}{\partial r} & 0 & \frac{\partial N_6}{\partial r} & 0 & \frac{\partial N_7}{\partial r} & 0 & \frac{\partial N_8}{\partial r} & 0 \\ \frac{N_1}{r} & 0 & \frac{N_2}{r} & 0 & \frac{N_3}{r} & 0 & \frac{N_4}{r} & 0 & \frac{N_5}{r} & 0 & \frac{N_6}{r} & 0 & \frac{N_7}{r} & 0 & \frac{N_8}{r} & 0 \\ 0 & \frac{\partial N_1}{\partial z} & 0 & \frac{\partial N_2}{\partial z} & 0 & \frac{\partial N_3}{\partial z} & 0 & \frac{\partial N_4}{\partial z} & 0 & \frac{\partial N_5}{\partial z} & 0 & \frac{\partial N_6}{\partial z} & 0 & \frac{\partial N_7}{\partial z} & 0 & \frac{\partial N_8}{\partial z} \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_1}{\partial r} & \frac{\partial N_2}{\partial z} & \frac{\partial N_2}{\partial r} & \frac{\partial N_3}{\partial z} & \frac{\partial N_3}{\partial r} & \frac{\partial N_4}{\partial z} & \frac{\partial N_4}{\partial r} & \frac{\partial N_5}{\partial z} & \frac{\partial N_5}{\partial r} & \frac{\partial N_6}{\partial z} & \frac{\partial N_6}{\partial r} & \frac{\partial N_7}{\partial z} & \frac{\partial N_7}{\partial r} & \frac{\partial N_8}{\partial z} & \frac{\partial N_8}{\partial r} \end{bmatrix}$$

To evaluate the derivative terms in $[B]$ the change of independent coordinate must be considered. This is accomplished by applying the chain rule and differentiating the shape functions with respect to the local coordinates β and α.

$$\frac{\partial N_i}{\partial \beta} = \frac{\partial N_i}{\partial r_i} \frac{\partial r_i}{\partial \beta} + \frac{\partial N_i}{\partial z_i} \frac{\partial z_i}{\partial \beta}$$

$$\frac{\partial N_i}{\partial \alpha} = \frac{\partial N_i}{\partial r_i} \frac{\partial r_i}{\partial \alpha} + \frac{\partial N_i}{\partial z_i} \frac{\partial z_i}{\partial \alpha}$$

In matrix notation, this becomes

$$\begin{Bmatrix} \frac{\partial N_i}{\partial \beta} \\ \frac{\partial N_i}{\partial \alpha} \end{Bmatrix} = \begin{bmatrix} \frac{\partial r}{\partial \beta} & \frac{\partial z}{\partial \beta} \\ \frac{\partial r}{\partial \alpha} & \frac{\partial z}{\partial \alpha} \end{bmatrix} \begin{Bmatrix} \frac{\partial N_i}{\partial r} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} \qquad (11)$$

The square matrix is defined as the Jacobian

$$[J] = \begin{bmatrix} \frac{\partial r}{\partial \beta} & \frac{\partial z}{\partial \beta} \\ \frac{\partial r}{\partial \alpha} & \frac{\partial z}{\partial \alpha} \end{bmatrix} \qquad (12)$$

The derivative in cylindrical coordinates can be determined as

$$\begin{Bmatrix} \frac{\partial N_i}{\partial r} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \beta} \\ \frac{\partial N_i}{\partial \alpha} \end{Bmatrix} \qquad (13)$$

The Jacobian is evaluated by differentiating equations 9.

In matrix notation this becomes

$$[J] = \begin{bmatrix} \frac{\partial N_1}{\partial \beta} & \frac{\partial N_2}{\partial \beta} & \frac{\partial N_3}{\partial \beta} & \frac{\partial N_4}{\partial \beta} & \frac{\partial N_5}{\partial \beta} & \frac{\partial N_6}{\partial \beta} & \frac{\partial N_7}{\partial \beta} & \frac{\partial N_8}{\partial \beta} \\ \frac{\partial N_1}{\partial \alpha} & \frac{\partial N_2}{\partial \alpha} & \frac{\partial N_3}{\partial \alpha} & \frac{\partial N_4}{\partial \alpha} & \frac{\partial N_5}{\partial \alpha} & \frac{\partial N_6}{\partial \alpha} & \frac{\partial N_7}{\partial \alpha} & \frac{\partial N_8}{\partial \alpha} \end{bmatrix} \begin{bmatrix} r_1 & z_1 \\ r_2 & z_2 \\ r_3 & z_3 \\ r_4 & z_4 \\ r_5 & z_5 \\ r_6 & z_6 \\ r_7 & z_7 \\ r_8 & z_8 \end{bmatrix} \qquad (14)$$

Substituting into equation (6) gives

$$\phi = \tfrac{1}{2} \iint \{ u \}^T [B]^T [D] [B] \{ u \} \, dVOL \qquad (15)$$

The displacement vectors are treated as constants, giving

$$\phi = \tfrac{1}{2} \{ u \}^T \iint [B]^T [D] [B] dVOL \{ u \} \qquad (16)$$

Castigliano's Theorem can now be applied giving:

$$\frac{\partial \phi}{\partial \{u\}} = \{ P \} = \iint [B]^T [D] [B] \, dVOL \, \{ u \} \quad (17)$$

By comparison with equation (3), the stiffness matrix is seen as

$$[K] = \iint [B]^T [D] [B] \, dVOL \quad (18)$$

If desired, displacements can be transformed to a different coordinate system by the equation

$$\{ \bar{u} \} = [T] \{ u \} \quad (19)$$

where $[T]$ is a coordinate transformation matrix.

The stiffness matrix then becomes

$$[K] = \iint [T]^T [B]^T [D] [B] [T] \, dVOL \quad (20)$$

In cylindrical coordinates, the volume integral can be written as

$$VOL = \iiint_{r\theta z} r \, dr \, d\theta \, dz = 2\pi \iint_{rz} r \, dr \, dz \quad (21)$$

where r is evaluated from equation (9).

This integral must be written in terms of the variables β and α.

The relation between the two coordinate systems is defined by Zienkiewicz[2] as

$$dr \, dz = | J | \, d\beta \, d\alpha \quad (22)$$

where $| J |$ is the determinant of the Jacobian matrix.

Making this change of integration variables, the stiffness matrix then becomes

$$[K] = 2\pi \iint [T]^T [B]^T [D] [B] [T] | J | \, r \, d\beta \, d\alpha \quad (23a)$$

The integration is best performed numerically using Gaussian quadrature. Then the stiffness matrix is written as

$$[K] = 2\pi \sum_{ij} [T]^T [B]^T [D] [B] [T] |J| \, r \, W_{ij} \qquad (23b)$$

where n is the order of the Gaussian integration required, and
W is a weighting factor.

As described by Janucik, a minimum integration order of 3 is recommended for this formulation.

B. THERMOELASTIC LOADS

Equivalent nodal forces are derived by assuming a virtual displacement at element nodes and equating the internal and external work. Let $\{ \zeta \}$ be a vector of virtual displacements at the nodes. The displacements and strains within the element then become

$$\{ u \} = [N] \{ \zeta \}$$
$$\{ \varepsilon \} = [B] \{ \zeta \} \qquad (24)$$

as defined by equations (8) and (10). If $\{ F \}$ is the vector of nodal forces, the external work is $\{ \zeta \}^T \{ F \}$. The internal work per unit volume done by stresses and distributed forces is

$$\{ \varepsilon \}^T \{ \sigma \} - \{ u \}^T \{ p \} \qquad (25)$$

where $\{ p \}$ is a vector of distributed forces;
Substituting from equation (24), this becomes

$$\{ \zeta \}^T [B]^T \{ \sigma \} - \{ \zeta \}^T [N]^T \{ p \} \qquad (26)$$

The total internal work is obtained by integrating this expression over the volume of the element. Integrating and equating the external and internal work gives

$$\{ \zeta \}^T \{ F \} = \{ \zeta \}^T < \int [B]^T \{ \sigma \} - [N]^T \{ p \} > dVOL \qquad (27)$$

Since this relation must be valid for any virtual displacement, the equivalent nodal forces become

$$\{ F \} = \int [B]^T \{ \sigma \} \, dVOL - \int [N]^T \{ p \} \, dVOL \qquad (28)$$

If initial stresses are specified , the equivalent nodal forces are

$$\{ F \} = \int [B]^T \{ \sigma \} \, dVOL \qquad (29)$$

If initial strains are specified instead of stresses, substitution of equation (4) into the above gives

$$\{ F \} = \int [B]^T [D] \{ \varepsilon_o \} \, dVOL \qquad (30)$$

Equivalent forces due to distributed loads are

$$\{ F \} = \int [N]^T \{ p \} \, dVOL \qquad (31)$$

The thermoelastic problem is treated as an initial strain case. The strain vector due to a temperature change DT is

$$\{ \varepsilon_o \} = \begin{Bmatrix} \alpha \, DT \\ \alpha \, DT \\ \alpha \, DT \\ 0 \end{Bmatrix} \qquad (32)$$

where α is the coefficient of thermal expansion.

Nodal forces due to a temperature change DT thus become

$$\{ F \} = \int [B]^T [D] \{ \varepsilon_o \} \, dVOL$$

or in cylindrical coordinates

$$\{ F \} = 2 \pi \iint [B] [D] \{ \varepsilon_o \} \, r \, dr \, dz \qquad (33)$$

Transforming to element coordinates and using Gaussian integration, this is expressed as

$$\{ F \} = 2 \pi \sum_{ij} \sum [B] [D] \{ \varepsilon_o \} |J| \, r \, W_{ij} \qquad (34)$$

C. STRESS RECOVERY

Stresses are calculated from equation (4),

$$\{ \sigma \} = [D] \{ \varepsilon \}$$

The strain vector must be relieved of all strains due to thermoelastic expansion since these are stress-free strains. The strain vector thus becomes

$$\{ \varepsilon \} = [B]^T \{ \zeta \} - \{ \varepsilon_o \} \qquad (35)$$

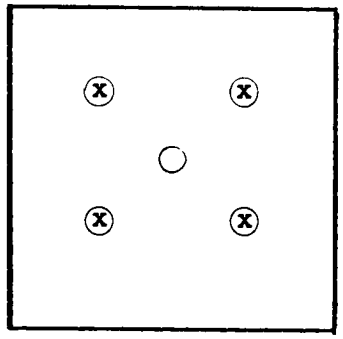
where $\{ \varepsilon_o \}$ is defined as in equation (32).

Stresses are evaluated at five points within each element -- the centroid and the four cornermost Gauss points as shown in Figure 2. Stresses are extrapolated linearly from the centroid through each cornermost Gauss point to each corner node. Stresses are output at the centroid and the four corner grids. At each stress point, stress invariants are calculated. These include the three principal stresses, the direction cosines associated with the first principal stress, the maximum shear, and the octahedral shear.

FIGURE 2. GAUSSIAN INTEGRATION AND STRESS RECOVERY POINTS

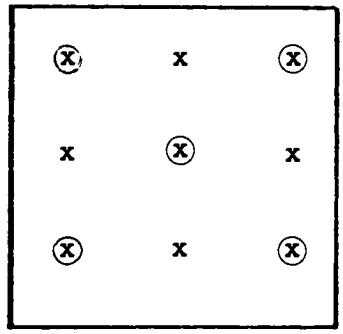
x - Gaussian integration points
◯ - Stress recovery points

Coordinates



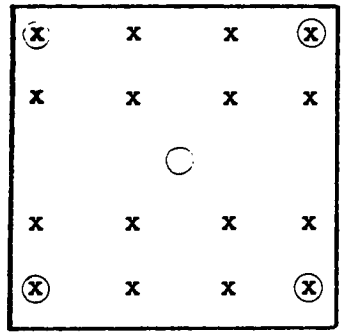
(±0.57735,±0.57735)

2 by 2 integration



(±0.77459,±0.77459)
(0.0 ,±0.77459)
(±0.77459, 0.0)
(0.0 , 0.0)

3 by 3 integration



(±0.86114,±0.86114)
(±0.86114,±0.33998)
(±0.33998,±0.86114)
(±0.33998,±0.33998)

4 by 4 integration

D. CONDUCTANCE MATRIX

The heat transfer equation is analagous to the structural equilibrium equation

$$[\,K\,]\,\{u\} = \{P\}$$

In structural analysis,

$[K]$ is the stiffness matrix,
$\{u\}$ is a vector of displacements, and
$\{P\}$ is a vector of applied loads.

In the thermal system, these are defined as:

$[K]$ is the heat conductance matrix,
$\{u\}$ is the nodal temperature vector, and
$\{P\}$ is a vector of applied heat flows ($P=qA$).

In heat transfer, only one degree of freedom -- the nodal temperature -- exists per grid point.

The thermal conductance matrix can be derived from a potential function in the same way that the stiffness matrix was derived from the strain energy. The thermal potential function is defined as

$$U = -\tfrac{1}{2} \int \tilde{q} \cdot \nabla u \; dVOL \qquad (36)$$

where $\tilde{q}$ the heat flux density, and
∇u the temperature gradient are

$$\begin{aligned} \tilde{q} &= q_1 i + q_2 j + q_3 k \\ \nabla u &= \frac{\partial u}{\partial x_1} i + \frac{\partial u}{\partial x_2} j + \frac{\partial u}{\partial x_3} k \end{aligned} \qquad (37)$$

Forming the dot product gives

$$\tilde{q} \cdot \nabla u = q_1 \frac{\partial u}{\partial x_1} + q_2 \frac{\partial u}{\partial x_2} + q_3 \frac{\partial u}{\partial x_3} \qquad (38)$$

The components of the flux q_i are related to the temperature gradient by

$$q_i = -\sum_j k_{ij} \frac{\partial u}{\partial x_j} \qquad (39)$$

where k_{ij} is a component of the material conductivity matrix, and
j is summed over the dimensions of the space.

Substituting equation (39) into equation (38) and expressing the result in matrix notation yields

$$U = \tfrac{1}{2} \int \left\{ \frac{\partial u}{x_i} \right\}^T \left[k_{ij} \right] \left\{ \frac{\partial u}{x_j} \right\} \, dVOL \qquad (40)$$

The temperatures are assumed to vary within the element by the isoparametric relation used before

$$u = \left[N \right] \{ u_o \} \qquad (41)$$

where $\{ u_o \}$ is a vector of constant nodal temperatures, and
$[N]$ is the matrix of isoparametric shape functions.

Since the nodal temperatures are constants, the derivative term is expressed as

$$\left\{ \frac{\partial u}{\partial x_i} \right\} = \left[\frac{\partial N}{\partial x_i} \right] \{ u_o \} \qquad (42)$$

Substituting this into equation (40) gives

$$U = \tfrac{1}{2} \int \{ u_o \} \left[\frac{\partial N}{\partial x_i} \right]^T \left[k_{ij} \right] \left[\frac{\partial N}{\partial x_j} \right] \{ u_o \} \, dVOL \qquad (43)$$

which is of the desired form

$$U = \tfrac{1}{2} \{ u_o \}^T \left[K \right] \{ u_o \} \qquad (44)$$

The conductance matrix $[K]$ is

$$\left[K \right] = \int \left[\frac{\partial N}{\partial x_i} \right]^T \left[k_{ij} \right] \left[\frac{\partial N}{\partial x_j} \right] \, dVOL \qquad (45)$$

In axisymmetric analysis, the radial and axial components of heat flow are considered. For homogeneous isotropic materials, the material matrix is

$$[k] = \begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix} \qquad (46)$$

The derivative matrix becomes

$$\left[\frac{\partial N}{\partial x_i}\right] = \begin{bmatrix} \frac{\partial N_1}{\partial r} & \frac{\partial N_2}{\partial r} & \frac{\partial N_3}{\partial r} & \frac{\partial N_4}{\partial r} & \frac{\partial N_5}{\partial r} & \frac{\partial N_6}{\partial r} & \frac{\partial N_7}{\partial r} & \frac{\partial N_8}{\partial r} \\ \frac{\partial N}{\partial z_1} & \frac{\partial N}{\partial z_2} & \frac{\partial N}{\partial z_3} & \frac{\partial N}{\partial z_4} & \frac{\partial N}{\partial z_5} & \frac{\partial N}{\partial z_6} & \frac{\partial N}{\partial z_7} & \frac{\partial N}{\partial z_8} \end{bmatrix} \qquad (47)$$

where as before

$$\begin{Bmatrix} \frac{\partial N_i}{\partial r} \\ \frac{\partial N_i}{\partial z} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \frac{\partial N_i}{\partial \beta} \\ \frac{\partial N_i}{\partial \alpha} \end{Bmatrix} \qquad (48)$$

where $[J]_{\beta,\alpha}$ is the Jacobian matrix defined in equation (14), and β, α are element local coordinates as shown in Figure 1.

Writing the volume integral as before and using Gaussian integration, the thermal conductance matrix is written as

$$[K] = \sum_I \sum_J \begin{bmatrix} \frac{\partial N_1}{\partial r} & \frac{\partial N_1}{\partial z} \\ \frac{\partial N_2}{\partial r} & \frac{\partial N_2}{\partial z} \\ \frac{\partial N_3}{\partial r} & \frac{\partial N_3}{\partial z} \\ \vdots & \vdots \\ \frac{\partial N_8}{\partial r} & \frac{\partial N_8}{\partial z} \end{bmatrix} \begin{bmatrix} k & 0 \\ 0 & k \end{bmatrix} \begin{bmatrix} \frac{\partial N_1}{\partial r} & \frac{\partial N_2}{\partial r} & \cdots & \frac{\partial N_8}{\partial r} \\ \frac{\partial N_1}{\partial z} & \frac{\partial N_2}{\partial z} & \cdots & \frac{\partial N_8}{\partial z} \end{bmatrix} W_{ij} |J| r \qquad (49)$$

In equation 49, i and j are summed over the order of the Gaussian integration. W_{ij} is a weighting function associated with the particular Gauss point.

V. ADDITIONS TO NASTRAN

NASTRAN has a capability for adding additional elements. This is most readily accomplished via the DUMMY element technique. Internal to NASTRAN exists a group of dummy subroutines. They are included only so that calls to these element dependent routines can be made. To implement a new element capability, functioning element routines must be written, compiled, and link edited into the NASTRAN program. Each of the routines must perform a specific task. Subroutine names include the identifying number of the DUMMY element (1-9). A subroutine KDUMi (i is the DUMMY element number) uses a NASTRAN table of element information to calculate the element stiffness or conductance matrix. Subroutine DUMi calculates equivalent element nodal loads due to thermal loading. Subroutine PDUMi sets up a connection array of grid points for plotting the undeformed and deformed structure. Two subroutines are used to calculate element stresses and a third is used to output the stress information in convenient formats. Subroutine SDUMi1 obtains element material data and calculates quantities which are constant for each element. This information is passed to subroutine SDUMi2 for final calculation of stresses and invariants. Headings and stress data are output in subroutine ODUMi. A listing of these subroutines is given in Appendix A.

There are a number of characteristics which are common to all the subroutines. First all subroutines are designed to perform calculations for one element at a time. The subroutines are called repeatedly by the

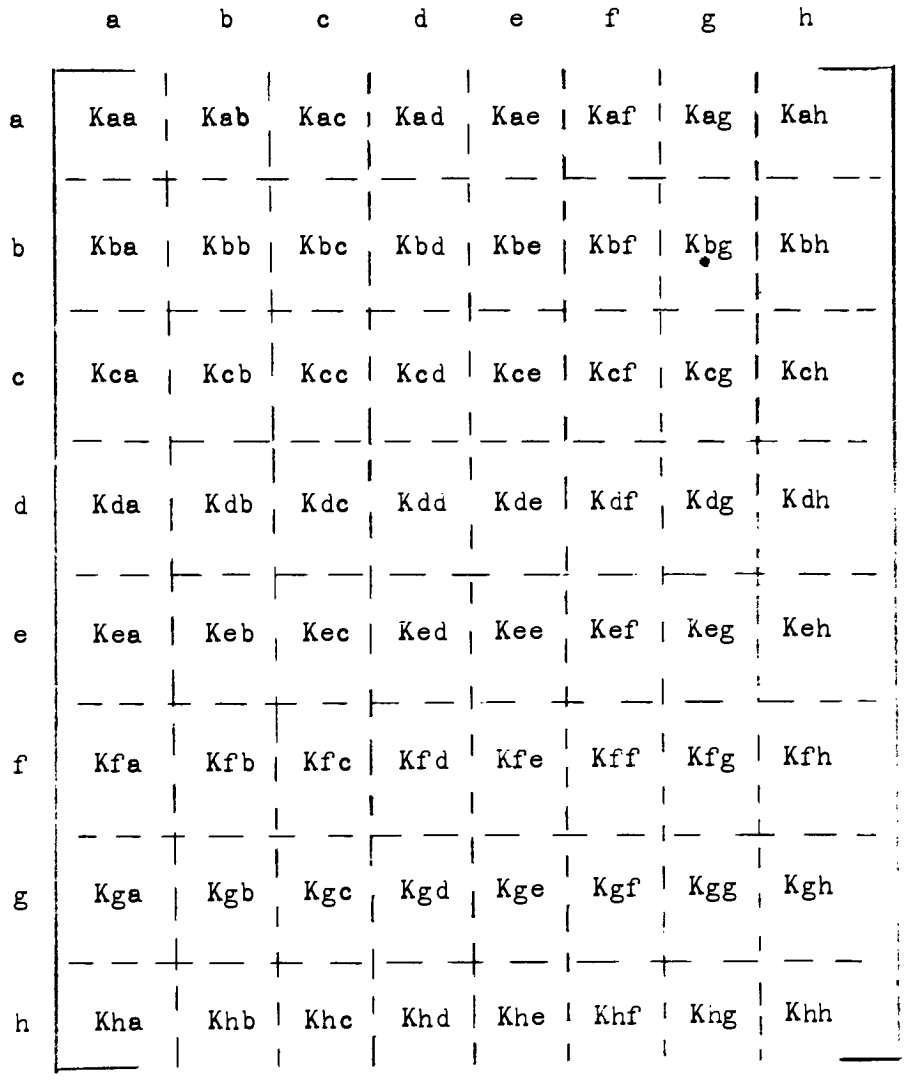
driving subroutine until all elements have been processed. Secondly all element information is passed to the new subroutines in labeled common areas. Element data is in regular NASTRAN table format but the amount of data is specified by the user as input data at execution time. The number of grid points connected, the amount of property data, and other connection data is specified on the ADUMi card. This data is used by the driving subroutines when preparing data for the element routines. These driving routines place the portion of the tables needed for the element currently being processed into the common areas for use by the element routines.

Finally a number of NASTRAN utility subroutines are available to the new subroutines. Some of the subroutines must be used to obtain material properties, temperatures, and displacements. Other subroutines can optionally be used for common matrix operations, such as multiplication and inversion, or to exercise other convenience options as desired. The utility subroutines and their functions are listed in Appendix A along with the element subroutines.

In NASTRAN all grid points are assumed to have 6 degrees of freedom (dof) -- 3 translations and 3 rotations -- except in heat transfer where only 1 dof per grid is defined. This convention allows connection of many diverse element types. The matrices generated in the element routines must associate 6 dof with each grid point. For the 8 noded CQDRNG8 element, only 2 dof per grid have stiffness associated with them. Thus the 16 by 16 full stiffness matrix (2 dof times 8 grids)

must be expanded to a 48 by 48 stiffness matrix (6 dof times 8 grids) by including null rows and columns before it can be added into the overall stiffness matrix for the structure. In the KDUMi routine, the full stiffness matrix is not calculated at one time. Instead NASTRAN uses the concept of a "pivot" grid. The element routine is called once for each grid in the element. The grid associated with the particular call to the element routine is called the "pivot" grid. The eight 6 by 6 matrix partitions, as shown in Figure 3, which connect the "pivot" grid to each of the eight element grids (including itself) are calculated and inserted into the overall stiffness matrix. This technique was used when NASTRAN was first developed because of the relatively simple elements available then. It was usually cheaper to recalculate portions of the element matrix than to store and retrieve it. The advent of higher order isoparametric elements and the numerical integration required for these elements made this an undesireable technique. A new technique in which the entire stiffness matrix is calculated and stored is currently being used for all new elements added to the MacNeal-Schwendler version of NASTRAN (MSC/NASTRAN). Since this newer technique is not now available for DUMMY elements, the older technique was used. The matrix generation time for this element is comparable to the times for existing elements. The new technique will be available for DUMMY elements in the future. The generation time for a CQDRNG8 element on an IBM 370/158 is approximately .30 seconds. Generation

FIGURE 3. STIFFNESS MATRIX PARTITIONS



times for similar elements range from .1 to .8 seconds.

Three new NASTRAN data cards are required for this element. These data cards (ADUM3, CQDRNG8, and PQDRNG8) are described in Appendix B. The new data cards define the properties and connectivity of the element. Existing NASTRAN data cards are used to locate grid points, constrain dof, apply loads, and define material properties.

VI. EXAMPLE PROBLEMS

The CQDRNG8 element has been implemented in version 34 of the MSC/NASTRAN finite element program. A number of classical problems have been solved using this element as implemented in NASTRAN. In all cases, the element showed good correlation with theoretical solutions. Three problems each will be used to demonstrate elastic and heat transfer capabilities. In addition to these theoretical test cases, an attempt was made to solve a thermoelastic problem previously solved using finite difference techniques. The example problems to be discussed are listed below in the order of presentation:

A. Bending of a circular simply supported plate.
Deflections are determined for a center load. Deflections and stresses are determined for a load on a concentric circular ring.

B. Thermal expansion of a circular disk.
Deflections are calculated for a disk which has expanded due to an increase in temperature.

C. Thermal expansion of a hollow sphere.
Deflections and stresses are calculated for two temperature distributions:
   1. a uniform temperature throughout the sphere which is higher than its reference temperature, and
   2. a linear temperature variation through the thickness of the sphere.

D. Heat flow in a circular disk.
A disk with a small center hole has the temperatures at its inner and outer surfaces constrained. Internal temperatures are determined using several finite element discretizations to demonstrate convergence. The same problem is also solved using an existing NASTRAN element to provide a comparison.

E. Heat flow in an axial rod.
Internal temperatures in a rod with constrained end temperatures are determined.

F. Heat flow in a hollow sphere.
Temperature distributions are determined for two sets of boundary conditions:
1. constrained temperatures at the inner and outer spherical surfaces, and
2. constrained temperatures at the inner spherical surface but convection at the outer surface.

G. Thermoelastic stresses in a hollow cylinder.
Stresses due to transient temperature distributions are determined for particular times and compared to a finite difference solution.

For convenience, computer results have been reduced to tabular and graphical form.

This element suffers the typical finite element problem with regard to stresses. Compatability of stresses in adjacent elements is not ensured by the formulation. Consequently when stresses are calculated for grid point locations which are common to several elements, different stresses result. The most common method of treating this discontinuity is to average each stress component at common grid points. The stresses presented in this thesis were calculated in this manner.

A. BENDING OF SIMPLY SUPPORTED FLAT PLATE

The thin circular flat plate of Figure 4 was analyzed using the finite element mesh shown. Theoretical results are from Timoshenko.[21] Two loading conditions were considered. First a load was applied at the center. Figure 4 is a graph of normalized deflection versus radial position. Results are presented for the theoretical solution, the CQDRNG8 element solution, and the CTRIAX6 element solution. The CTRIAX6 is the triangular counterpart of the CQDRNG8 and currently exists in NASTRAN. It should be noted that the theoretical solution is for bending deflection only whereas the finite element solutions include the effects of shear. Tabular results are presented in Table 1. The finite element solutions are nearly identical and are within 2.5 % of theory.

The second loading condition considered is that of a uniform load applied along a concentric circle. Displacement results are presented graphically in Figure 5 and in tabular form in Table 2. Again the displacements agree very well with the theoretical result from Timoshenko. The maximum difference was less than 1.25 %. Stresses were also calculated for this case. Stress results are given in Figure 6 and Table 3. Both the radial and hoop stresses agree to within 11.1 % of theoretical results.

FIGURE 4

Deflections of simply supported plate with center load.

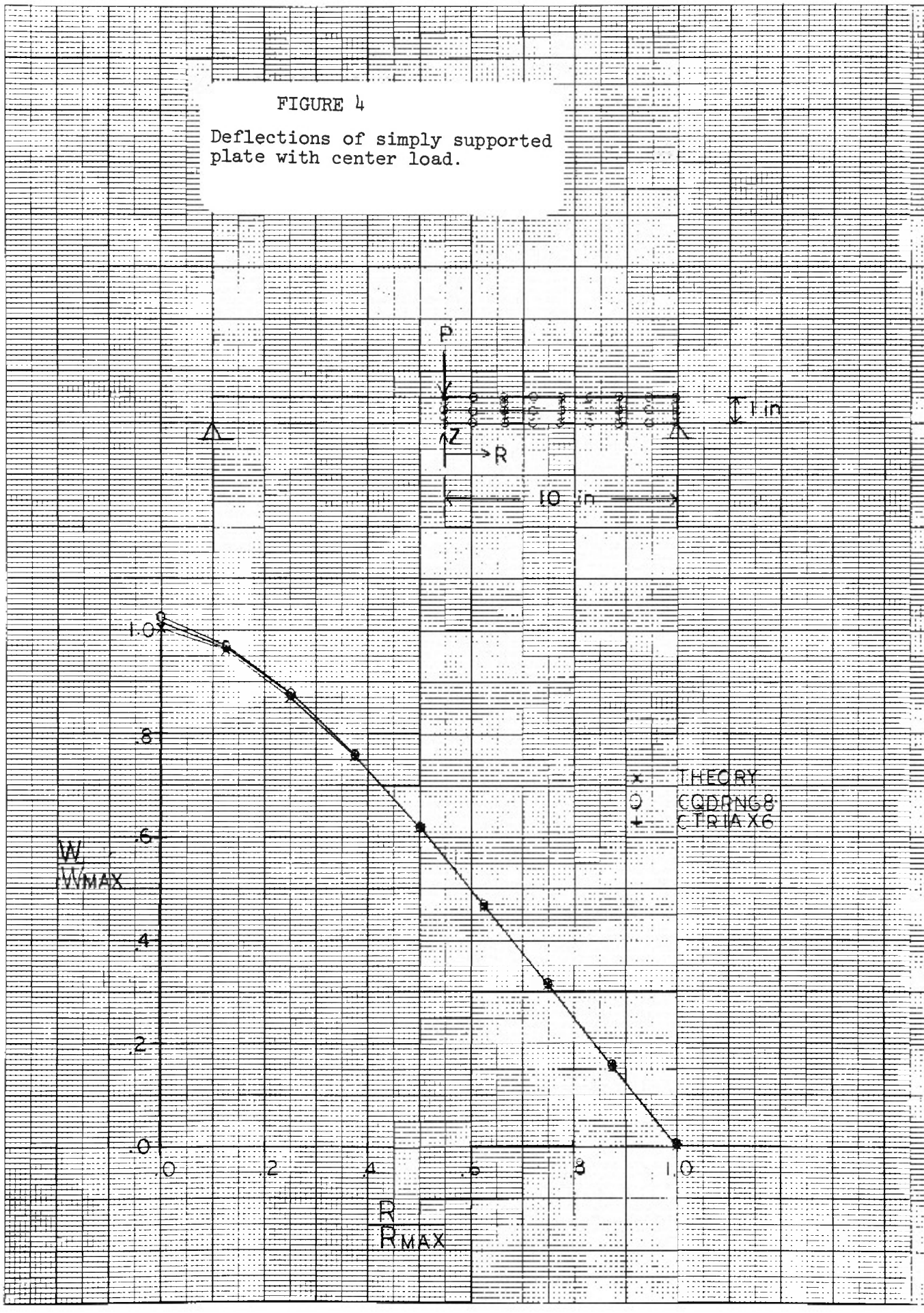
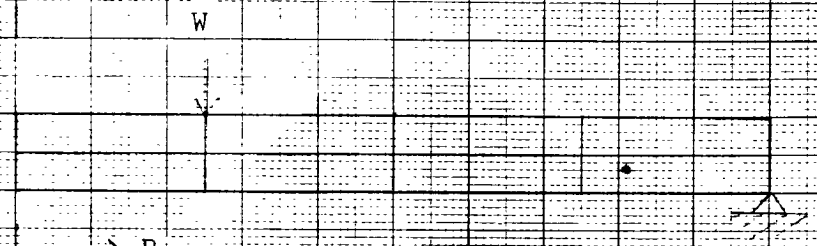


TABLE 1. Deflections of simply supported plate with center load

RADIUS	DEFLECTIONS (10^{-4} inches)			
	CQDRNG8	THEORY	% DIFFERENCE	CTRIAX6
0.0	1.925	1.879	2.41	1.913
1.25	1.824	1.802	1.21	1.814
2.50	1.648	1.635	.79	1.644
3.75	1.421	1.413	.61	1.418
5.00	1.161	1.155	.53	1.159
6.25	.880	.876	.48	.878
7.50	.588	.585	.47	.587
8.75	.292	.291	.59	.291
10.00	.0	.0	.0	.0

FIGURE 5

Deflections of simply supported plate with concentric circular load.



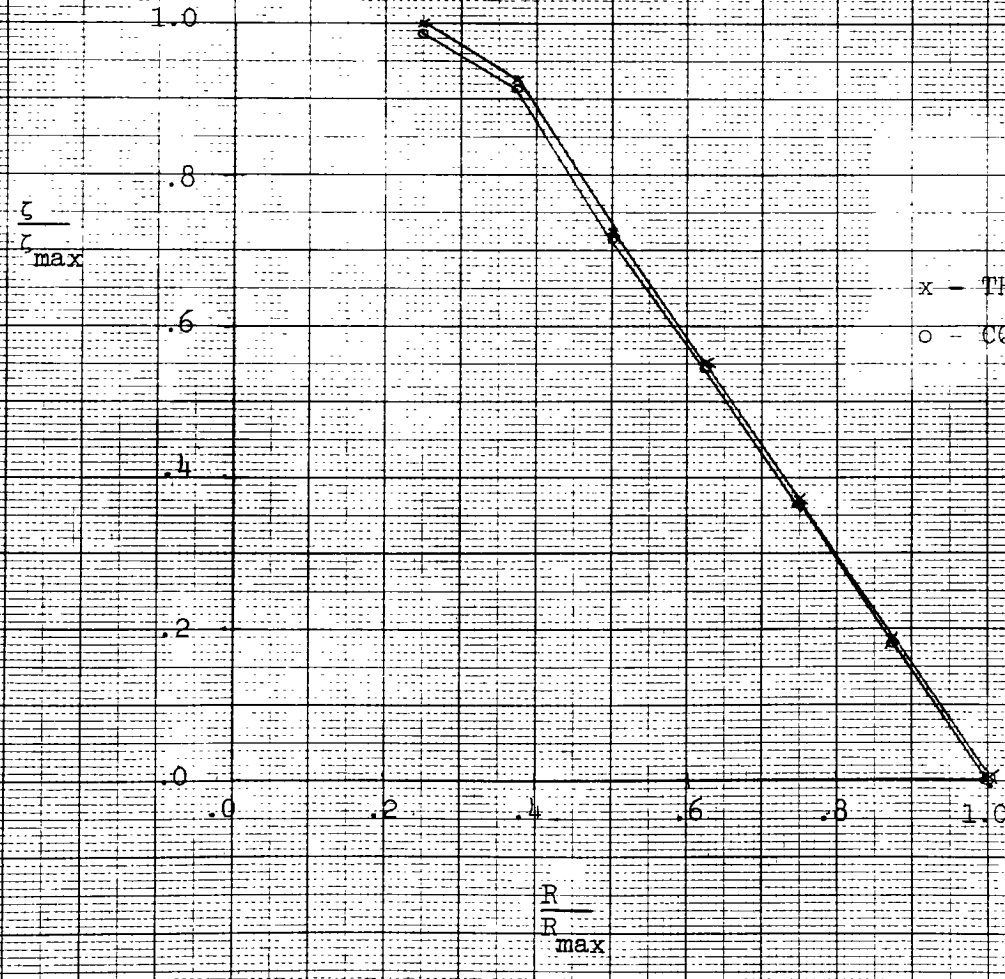


TABLE 2. DEFLECTIONS OF SIMPLY SUPPORTED CIRCULAR PLATE WITH LOAD ON CONCENTRIC CIRCLE

Radius	Deflections (10^{-3} inches)		
	CQDRNG8	Theory	% Difference
12.5	3.664	3.709	1.21
18.75	3.217	3.252	1.08
25.0	2.658	2.681	.86
31.25	2.026	2.043	.83
37.5	1.358	1.368	.73
43.75	.676	.681	.73
50.0	.0	.0	.0

FIGURE 6

Stresses in simply supported circular plate with load on concentric circle

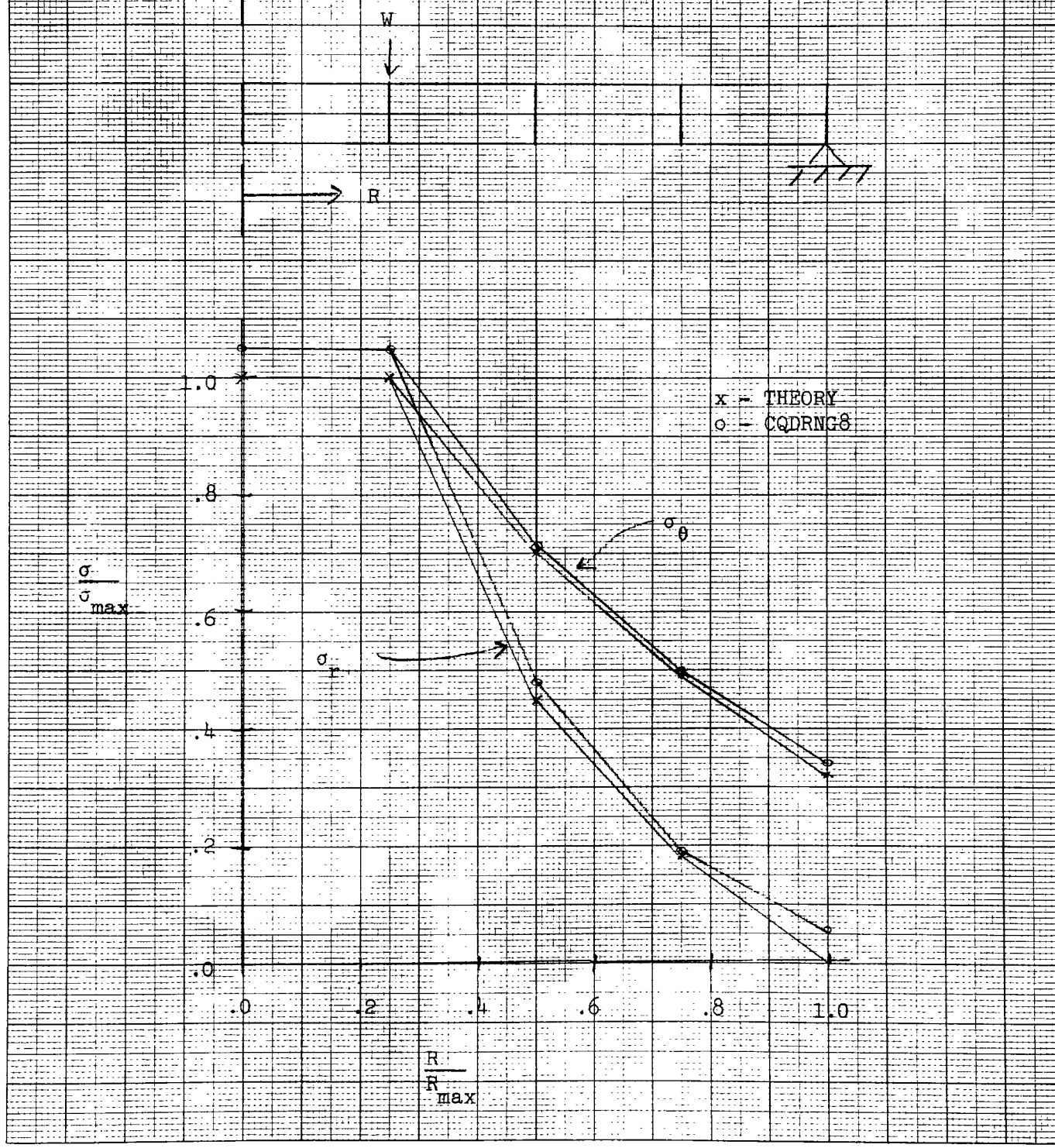


TABLE 3 - Stresses in simply supported circular plate with concentric loading

RADIUS	σ_R			σ_θ		
	CQDRNG8	THEORY	% DIFF	CQDRNG8	THEORY	% DIFF
12.5	105.	101.	3.96	105.	101.	3.96
25.0	48.	46.	4.35	72.	71.	1.41
37.5	20.	18.	11.11	50.	49.	2.04
50.	6.	0.	-	35.	32.	9.38

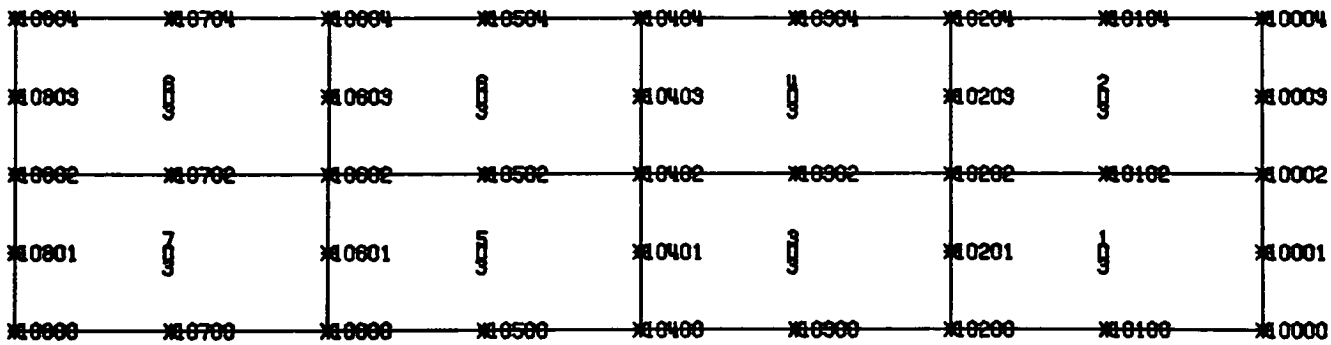
B . THERMAL EXPANSION OF CIRCULAR DISK

To test the thermoelastic loading capability, a circular disk with a center hole was subjected to a uniform temperature change. The temperature of the ring was raised by 100 °F resulting in a thermoelastic expansion of the disk. Figure 7 is a plot of the finite element model. Grids are indicated with an * . Grid identification numbers are to the right of the *'s. In the center of each element is the element identification number. The disk was unconstrained in the radial direction and was constrained at the outside in the axial direction. As shown in Timoshenko, the disk should undergo a stress free radial and axial growth. Displacement results are shown in Figure 8 and Table 4. The predicted displacements are exactly those from theory (0.0 % error). As expected all stresses were zero.

FIGURE 7

Plot of circular disk model.

1 4/18/77

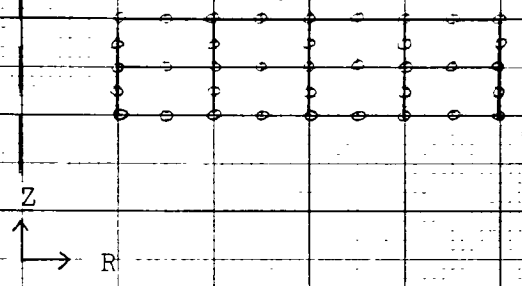


FLAT CIRCULAR RING

UNDEFORMED SHAPE

FIGURE 8

Deflections of disk with center hole - uniform temperature change.



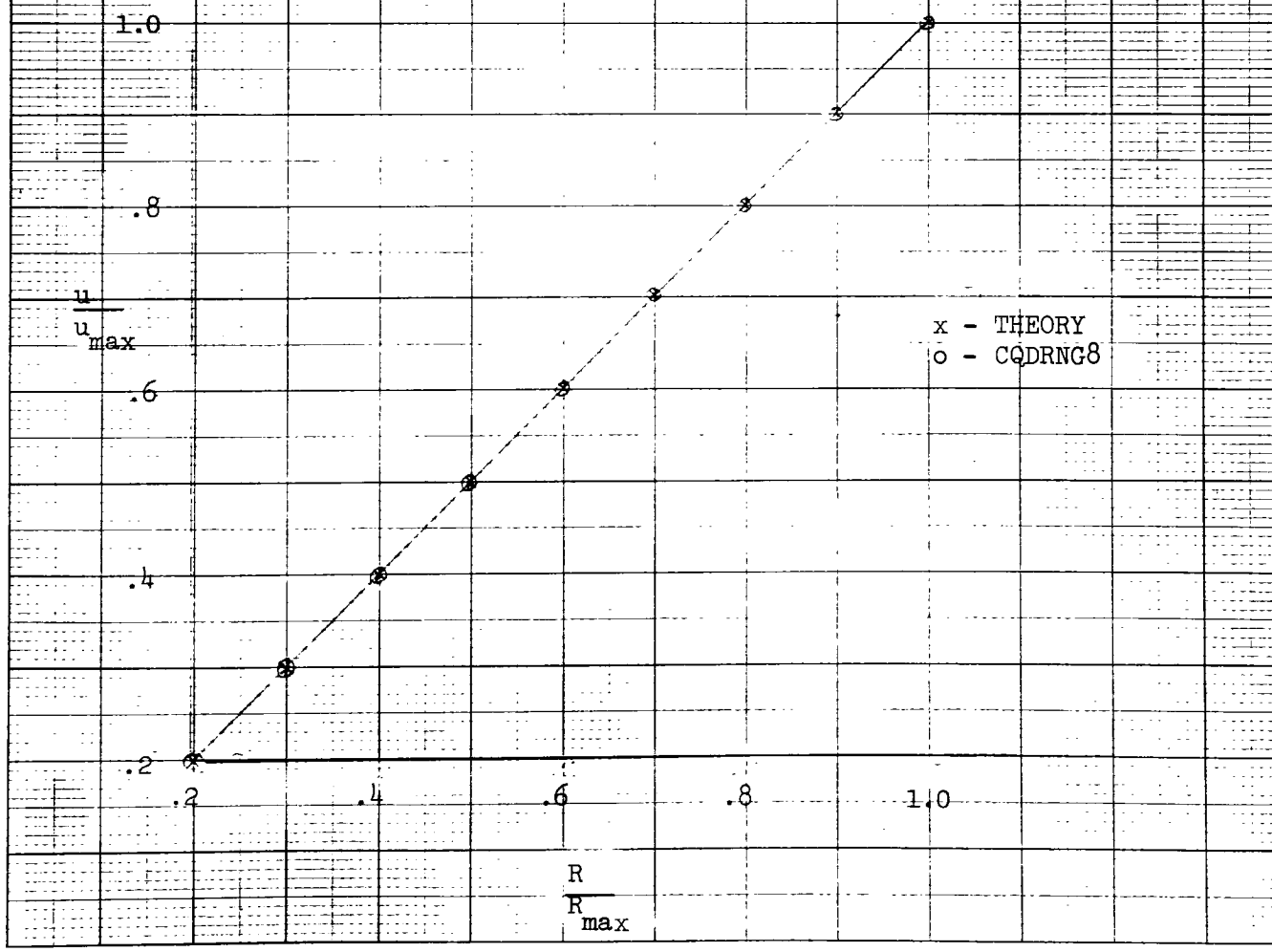


TABLE 4 - Thermal deflections in disk

RADIUS	DEFLECTIONS		
	CQDRNG8	THEORY	% DIFFERENCE
10.0	.0010	.0010	0.0
15.0	.0015	.0015	0.0
20.0	.0020	.0020	0.0
25.0	.0025	.0025	0.0
30.0	.0030	.0030	0.0
35.0	.0035	.0035	0.0
40.0	.0040	.0040	0.0
45.0	.0045	.0045	0.0
50.0	.0050	.0050	0.0

C. THERMAL EXPANSION OF HOLLOW SPHERE

A symmetric section of a hollow sphere was modeled as shown in the plot of Figure 9. This sphere was subjected to two different temperature distributions. Symmetric boundary conditions were applied to the cuts of symmetry. Theoretical results from Timoshenko are given.

In the first case, the sphere is at a uniform temperature which is 100°F above its reference temperature. The sphere expands radially due to the new temperature distribution. As shown in Figure 10 and Table 5, the CQDRNG8 displacements are exactly those expected (0.0 % error). Since the sphere is unconstrained, all stresses are zero.

The second case considered a linear variation of temperature through the thickness. This temperature distribution caused internal bending stresses. Theoretical bending stresses at the surface for a 100°F temperature difference are 2080 psi. The CQDRNG8 predicted stresses of 1950 psi at the OD and 2250 at the ID. These predicted stresses are within 8 % of theory.

FIGURE 9

Plot of hollow sphere model

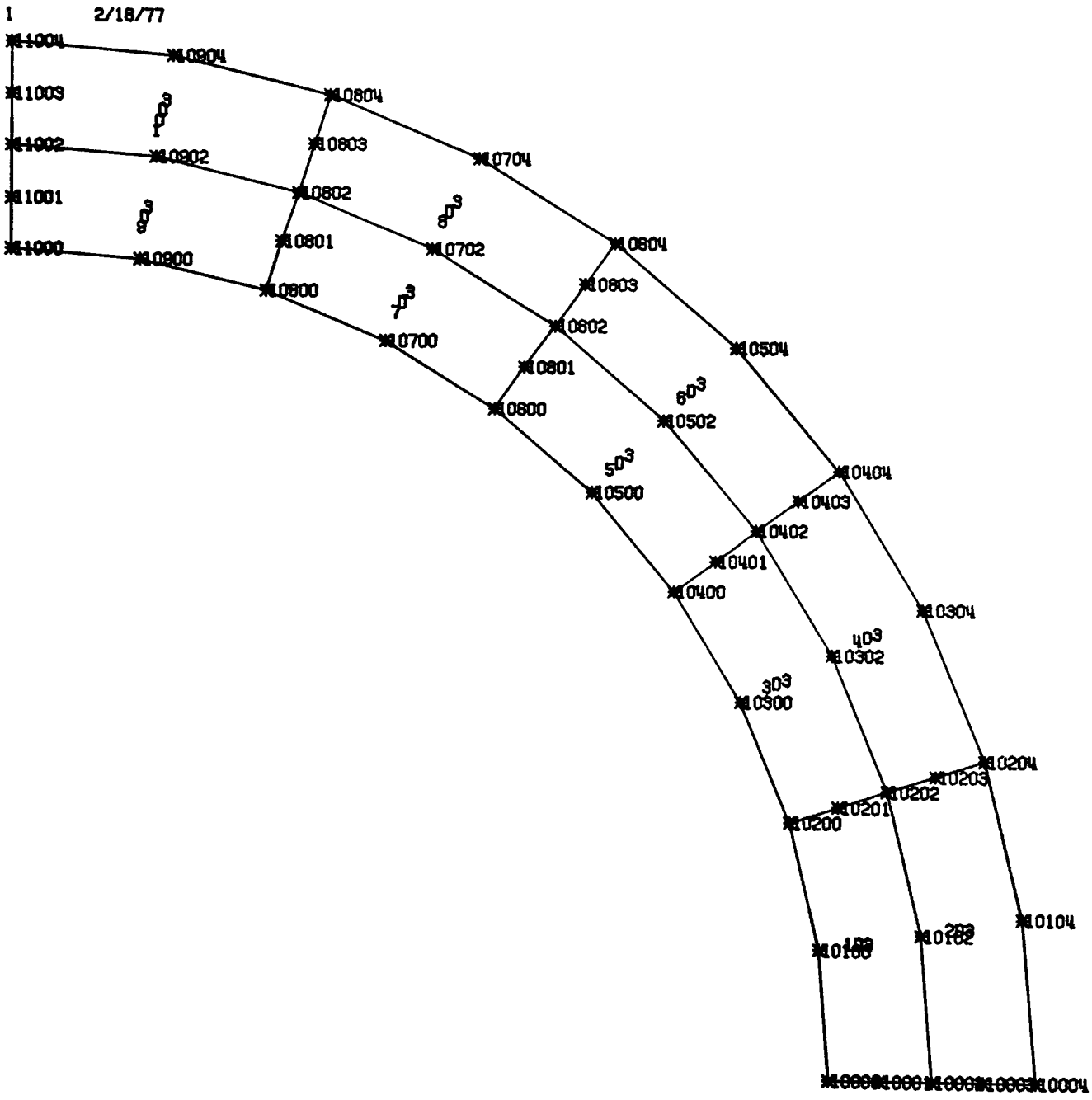
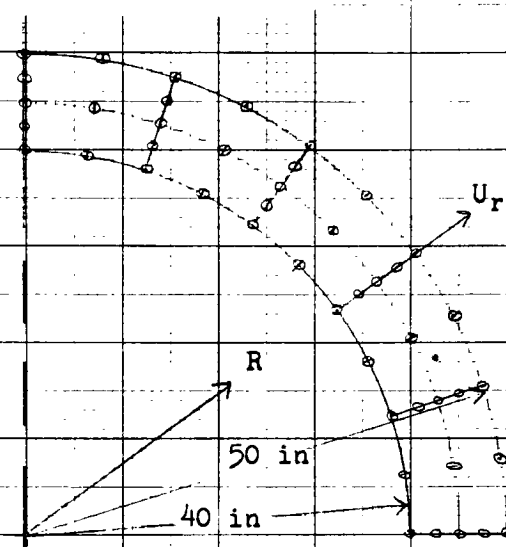


FIGURE 10

Thermal expansion of sphere



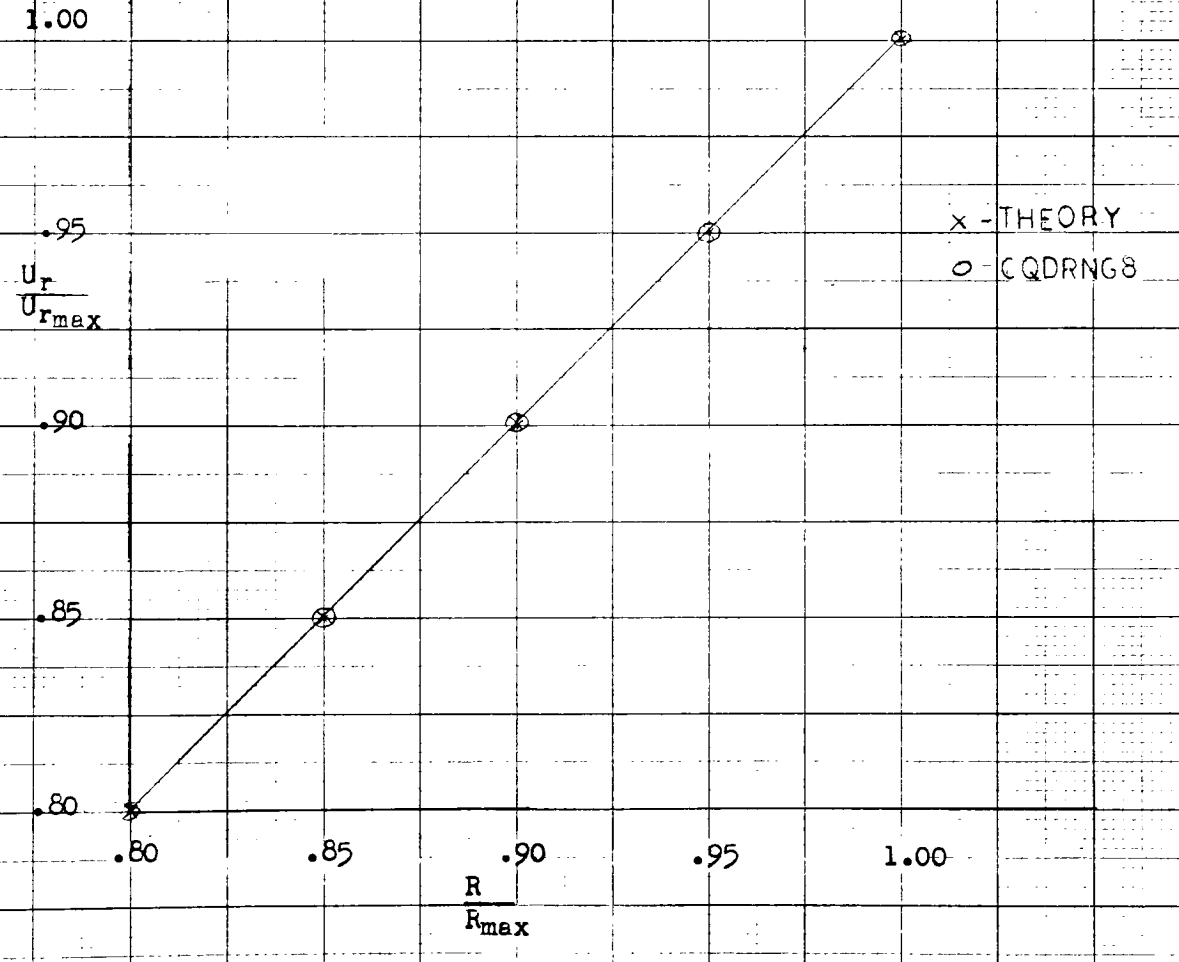


TABLE 5 - Thermal expansion of hollow sphere

RADIUS	RADIAL DEFLECTION		
	CQDRNG8	THEORY	% DIFFERENCE
40.0	.00400	.00400	0.0
42.5	.00425	.00425	0.0
45.0	.00450	.00450	0.0
47.5	.00475	.00475	0.0
50.0	.00500	.00500	0.0

D. HEAT FLOW IN A CIRCULAR DISK

To demonstrate the convergence of the CQDRNG8 element, the problem of heat flow in a circular disk was solved. The disk has a small centerhole constrained to 0.0°F. The outer diameter of the disk is constrained to 100.0°F. This problem was solved using the three different discretizations shown in Figure 11. The models contained 8, 2, and 1 CQDRNG8 elements respectively. The problem was also solved using three similar gridworks of CTRIARG elements. These models are shown in Figure 12. The CTRIARG is a three noded triangular ring element which currently exists in NASTRAN. Each CQDRNG8 element has been filled with eight CTRIARG elements and an additional degree of freedom has been added at the center. The CTRIARG element has a linear temperature variation as compared to the quadratic variation of the CQDRNG8 element.

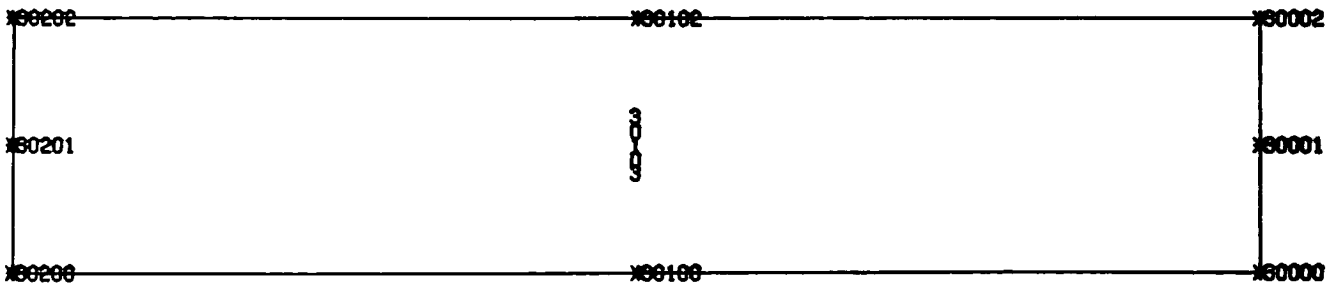
The results of this study are shown in Figure 13 and Table 6. Theoretical results are from Carslaw.[22] For the coarsest model, the 1 CQDRNG8 element with 2 unconstrained degrees of freedom gives the same results as 8 CTRIARG elements having 3 dof. The temperatures are in error by 10.57 %. The 2 element CQDRNG8 model with 7 unconstrained dof predicts temperatures which are in error by 1.58 to 12.16 %. The CTRIARG model with 16 elements and 9 unconstrained dof predicts temperatures greater in error. The error in the CTRIARG model ranges from 2.23 to 13.70 %. In the finest models, the CQDRNG8 with 8 elements and 27 unconstrained dof has a maximum error of 9.96 %. The CTRIARG model with 64 elements and 35 unconstrained dof had a maximum error of 12.33 %.

In all cases, the CQDRNG8 models had one-eighth the number of elements and at least one-fourth fewer degrees of freedom than the CTRIARG models. For the coarest model, the single CQDRNG8 element was comparable to the eight CTRIARG elements. For the other models, a moderate increase in accuracy was realized. The CQDRNG8 element does exhibit the convergence characteristic. Each refinement of the model resulted in improved accuracy at corresponding locations.

FIGURE 11 a

Circular disk model
1 CQDRNG8 element
2 unconstrained dof

3 4/28/77 3

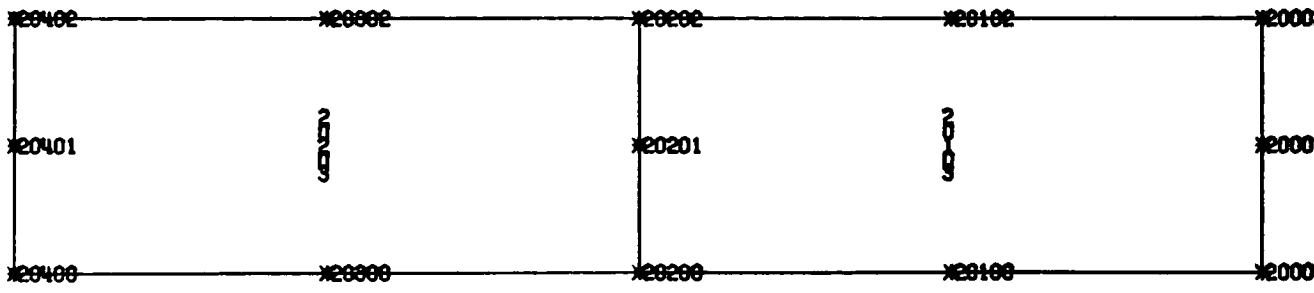


CQDRNG8 THERMAL TEST
ENFORCED CUBAR DISP
ENFORCED TEMPS AT CENTER AND OD
UNDEFORMED SHAPE

FIGURE 11 b

Circular disk model
2 CQDRNG8 elements
7 unconstrained dof

2 4/28/77 2

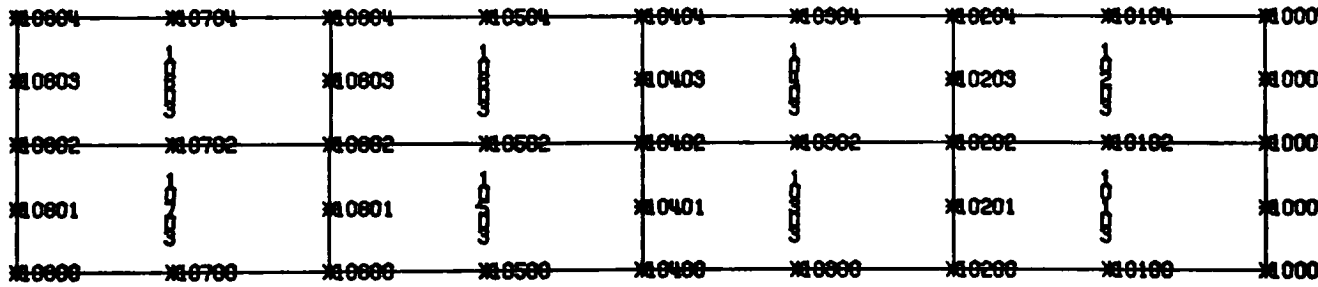


CQDRNG8 THERMAL TEST
ENFORCED TEMPS AT CENTER AND OD
UNDEFORMED SHAPE

FIGURE 11 c

Circular disk model
8 CQDRNG8 elements
27 unconstrained dof

1 4/28/77 1

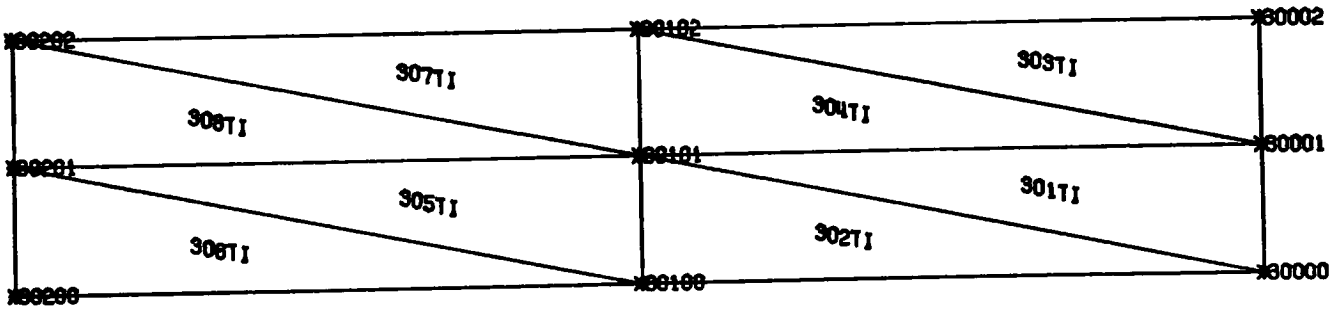


CQDRNG8 THERMAL TEST
FEATURE CIRCULAR DISK
ENFORCED TEMPS AT CENTER AND OD
UNDEFORMED SHAPE

FIGURE 12 a

Circular disk model
8 CTRIARG elements
3 unconstrained dof

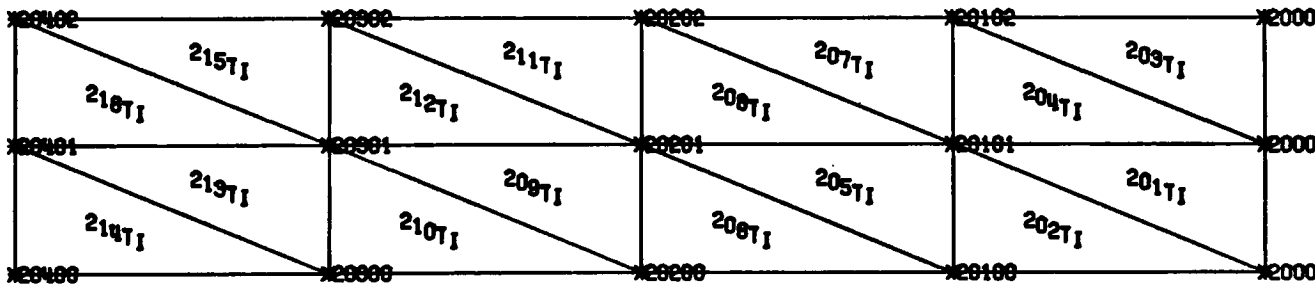
3 4/28/77 3



CTRIARG MODEL
FLAT CIRCULAR DISK
ENFORCED TEMPS AT CENTER AND OD
UNDEFORMED SHAPE

FIGURE 12 b
Circular disk model
16 CTRIARG elements
9 unconstrained dof

2 4/28/77 2

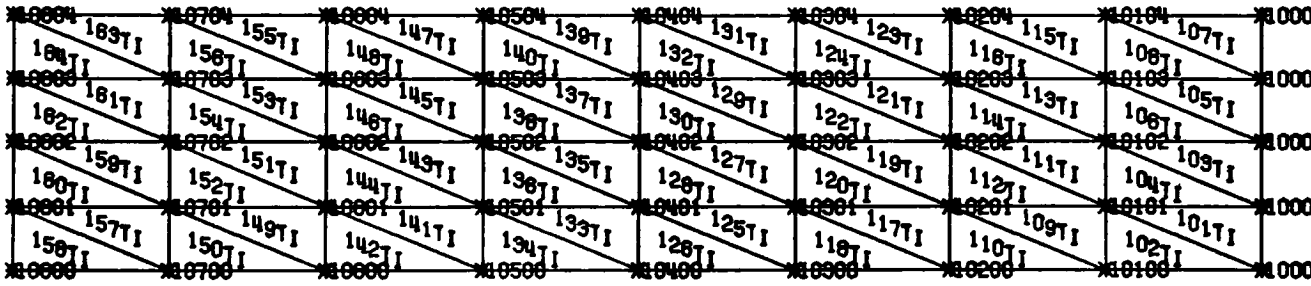


CTRIARG MODEL
FLAT CIRCULAR DISK
ENFORCED TEMPS AT CENTER AND OD
UNDEFORMED SHAPE

FIGURE 12 c

Circular disk model
64 CTRIARG elements
35 unconstrained dof

1 4/27/77 1



CTRIARG MODEL
FLAT CIRCULAR DISK
ENFORCED TEMPS AT CENTER AND 60
UNDEFORMED SHAPE

FIGURE 13

Circular disk temperatures
Convergence test

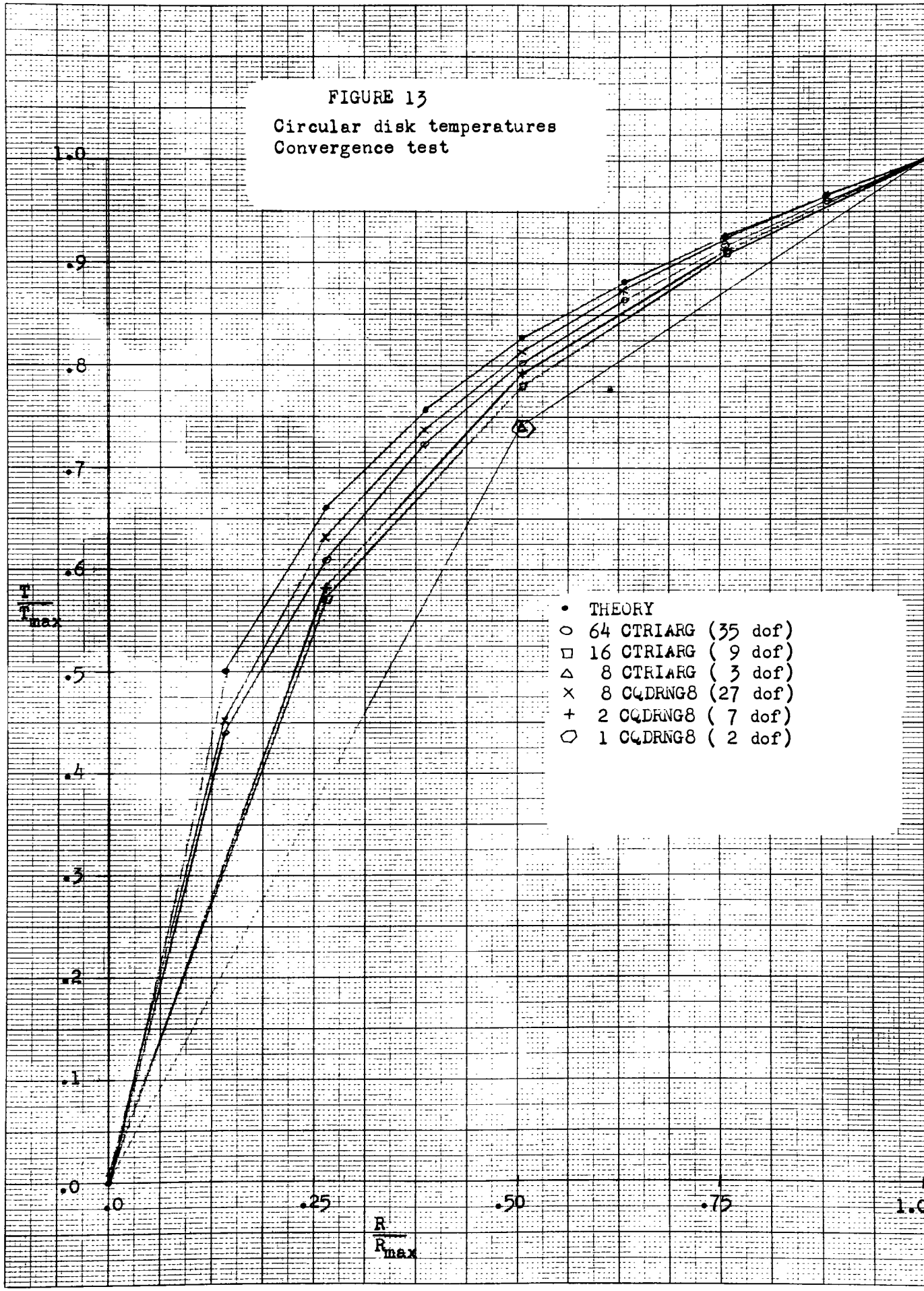


TABLE 6 - Circular disk temperatures - convergence test

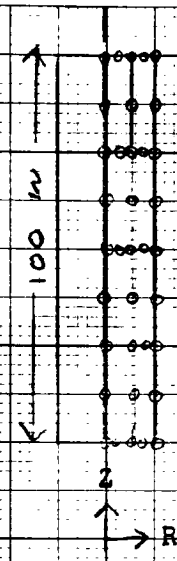
RADIUS	TEMPERATURES THEORY	CTRIARG RESULTS 35 dof	% Diff	9 dof	% Diff	3 dof	% Diff
1.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7.125	50.1942	44.00	12.33				
13.250	66.0527	61.51	6.86	57.00	13.70		
19.375	75.7660	72.46	4.36				
25.500	82.7878	80.41	2.86	77.93	5.86	74.03	10.57
31.625	88.2906	86.67	1.83				
37.750	92.8161	91.82	1.07	90.75	2.13		
43.875	96.6246	96.19	.45				
50.000	100.000	100.00	0.0	100.00	0.0	100.00	0.0

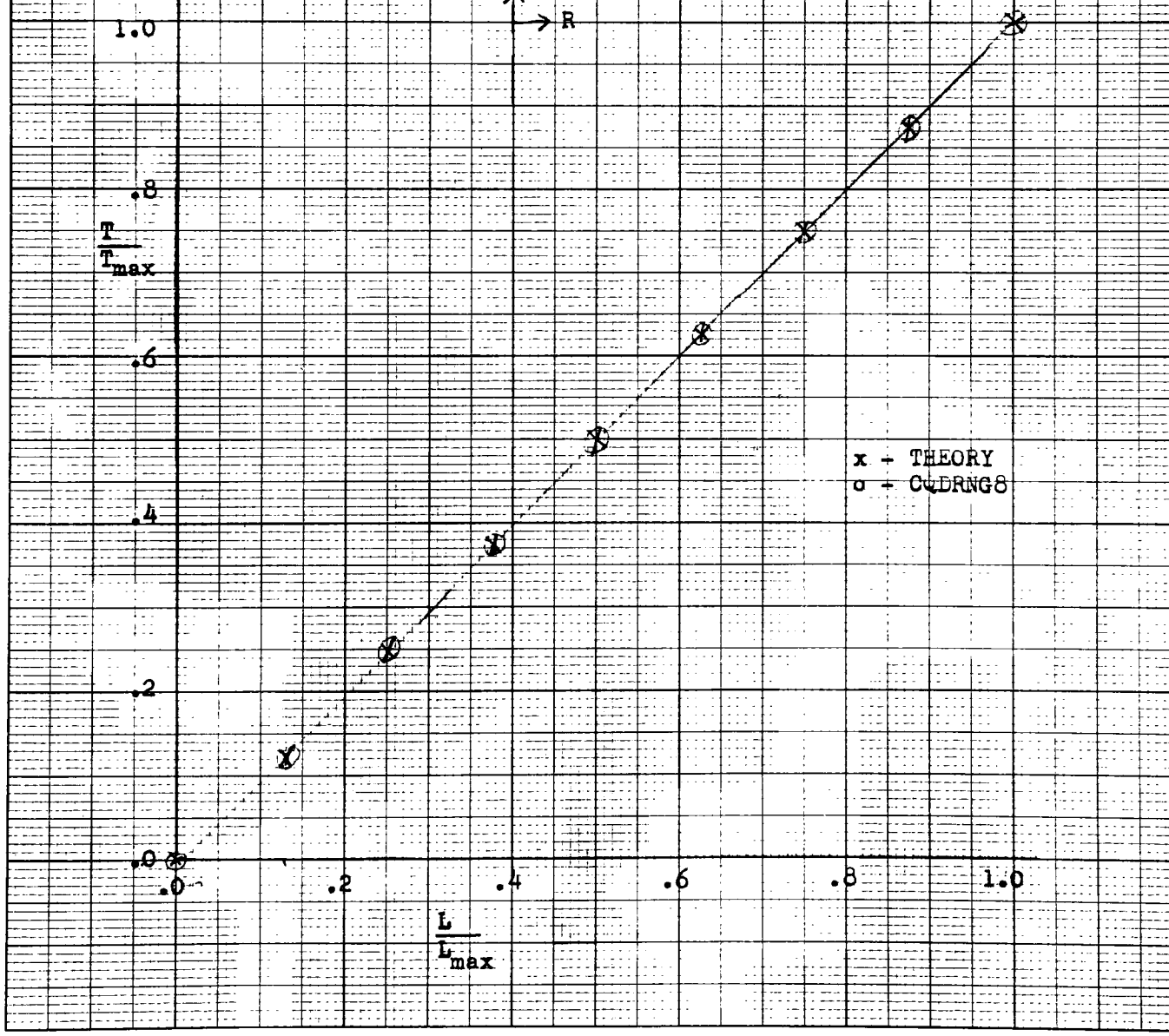
RADIUS	TEMPERATURES THEORY	CQDRNG8 RESULTS 27 dof	% Diff	7 dof	% Diff	2 dof	% Diff
1.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7.125	50.1942	45.19	9.96				
13.250	66.0527	63.22	4.28	58.02	12.16		
19.375	75.7660	73.71	2.71				
25.500	82.7878	81.34	1.74	79.36	4.13	74.02	10.57
31.625	88.2906	87.30	1.12				
37.750	92.8161	92.21	.65	91.35	1.58		
43.875	96.6246	96.38	.25				
50.000	100.00	100.00	.0	100.00	.0	100.00	.0

E. HEAT FLOW IN AN AXIAL ROD

Temperatures were determined in an axial rod with constrained end temperatures. The theoretical solution from Carslaw predicts a linear temperature variation through the rod. The model and results are shown in Figure 14. The CQDRNG8 model predicts the theoretical results exactly.

FIGURE 14
Temperatures in axial rod
Enforced end temperatures





F. HEAT FLOW IN A HOLLOW SPHERE

The final thermal test case is that of a hollow sphere. The sphere is analyzed for two sets of boundary conditions. The first case is that of constrained temperatures at both the inner and outer diameters. The theoretical solution is presented in Carslaw. The finite element model is the same as that shown in Figure 9. Figure 15 and Table 7 show both the theory and CQDRNG8 results. For this case, the CQDRNG8 results are exact (0.0 % error).

For the second case, the inner diameter temperatures are constrained and the outer diameter is subjected to convection. To represent the convection boundary condition, special NASTRAN heat transfer boundary condition elements (CHBDY) must be used in conjunction with the CQDRNG8 elements. The theory for this case is also found in Carslaw. Figure 16 and Table 8 show the comparison between theory and CQDRNG8 results. Results agree to within 2.0 %.

FIGURE 15

Temperature in hollow sphere
Enforced temperature at surfaces

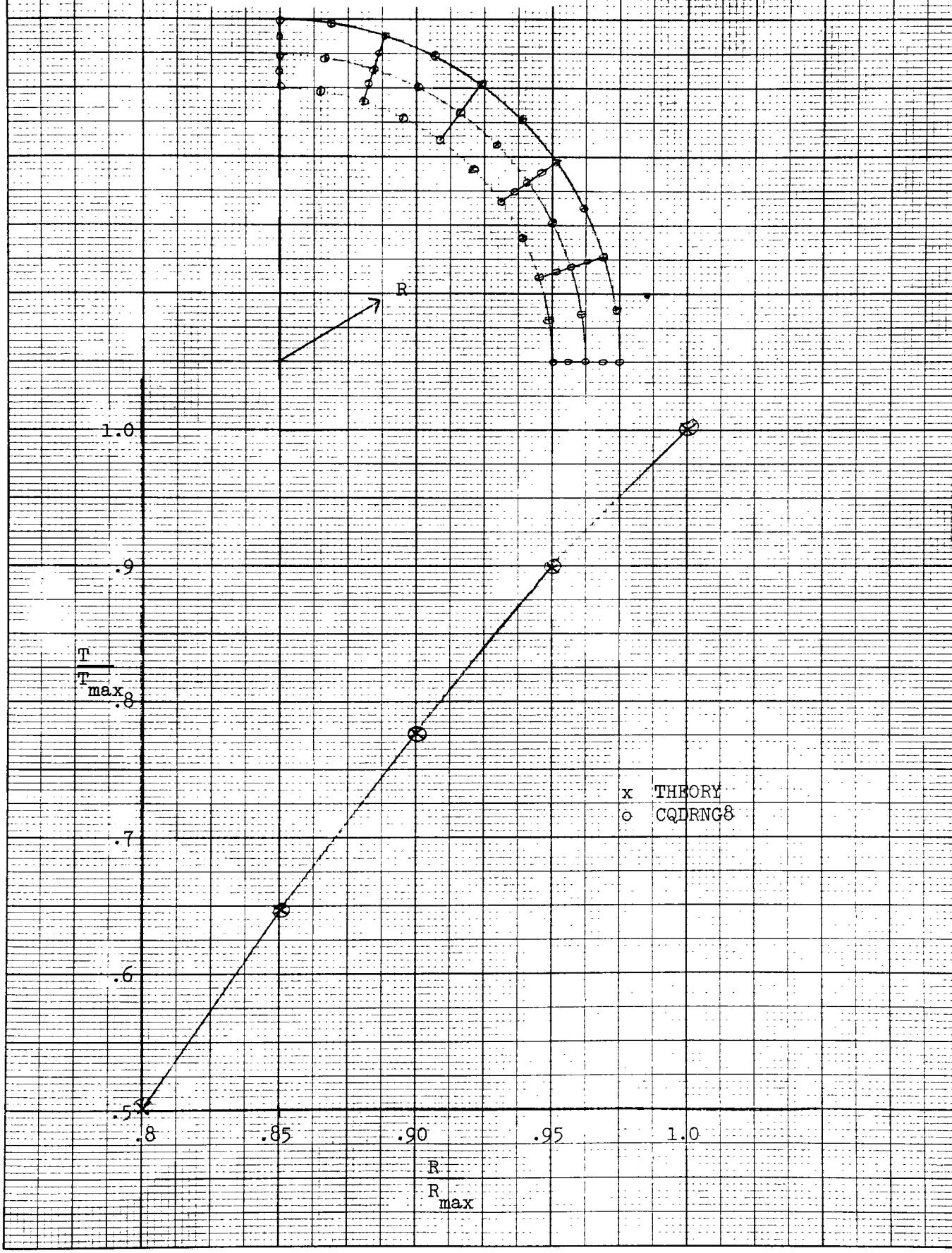
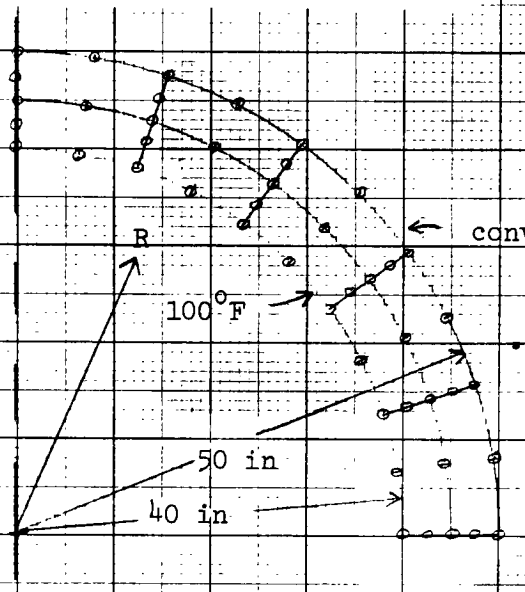


TABLE 7 - Temperatures in hollow sphere
Enforced temperatures at surfaces

RADIUS	TEMPERATURES		
	CQDRNG8	THEORY	% DIFFERENCE
40.0	100.00	100.00	0.0
42.5	129.41	129.41	0.0
45.0	155.56	155.56	0.0
47.5	178.95	178.95	0.0
50.0	200.00	200.00	0.0

FIGURE 16

Temperatures in hollow sphere
Enforced temperatures at ID
Convection at OD



1.0
.9
$\frac{T}{T_{max}}$
.8
.7
.6
.5

x THEORY
o CQDRNG8

.80 .85 .90 .95 1.00

$\frac{R}{R_{max}}$

TABLE 8 - Temperatures in hollow sphere
Enforced temperature at ID
Convection at OD

RADIUS	TEMPERATURES		
	CQDRNG8	THEORY	% DIFFERENCE
40.0	100.00	100.00	.0
42.5	127.97	128.28	.24
45.0	152.84	153.42	.38
47.5	177.05	175.91	.65
50.0	199.04	196.15	1.47

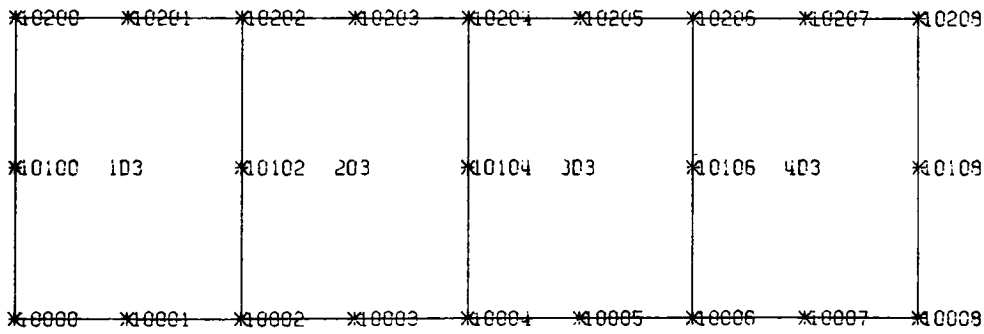
G. THERMOELASTIC STRESSES IN A HOLLOW CYLINDER

Schlottner[23] presented a finite difference solution for transient thermal stresses in a hollow cylinder. The cylinder, at a uniform initial temperature and with convection at the outer diameter, was subjected to an increase in the ambient temperature. The resulting transient temperature distributions create internal stresses. Since the CQDRNG8 element does not include a transient heat transfer capability, the heat transfer portion of the problem could not be solved. However thermal stresses could be determined at particular time steps if a reasonable estimate of the temperature distribution could be made. Schlottner's solution gives the temperatures at the ID and OD of the hollow cylinder at various times. For a first approximation, the temperature distribution through the cylinder was assumed to be linear. The finite element model is shown in Figure 17. As shown in Table 9, , the stresses for the two time steps listed are approximately 10.0 to 25.0 % lower than those predicted by Schlottner. A second approximation of the temperature distribution was arrived at by solving the steady state heat problem using boundary conditions from Schlottner's solution. At the 26^{th} time step of Schlottner's solution, the OD temperature was close to a steady state condition but the ID temperature was not. For the steady state problem solved, the ID temperature was constrained to the value predicted by Schlottner and the convection boundary condition at the OD was retained. As

FIGURE 17

Plot of hollow cylinder model .

1 4/12/77 1



THERMAL STRESSES IN THICK CYLINDER - ID=6.0, OD=24.0
ASSUMED LINEAR TEMP DIST FROM SCHLOTTNER EX 2.
UNDEFORMED SHAPE

TABLE 9 - Stresses in hollow cylinder

TIME STEP	SCHLOTTNER STRESS		LINEAR TEMP STRESS		% DIFF		SS TEMP STRESS		% DIFF	
	ID	OD	ID	OD	ID	OD	ID	OD	ID	OD
15	16000	19000	12000	14000	25.0	26.3				
26	11000	11000	10000	8000	9.1	27.3	12600	6000	14.5	45.5

shown in Table 9, this resulted in poorer correlation at the ID but much poorer correlation at the OD.

This comparison is by no means a rigorous test of the CQDRNG8 element's capabilities and is included only to provide a comparison to another numerical method. To determine the correct thermoelastic stresses, the correct temperature distribution must be known. Considering the approximate nature of the temperature assumption, the predicted stresses are quite reasonable.

VII. SUMMARY

The development and application of an axisymmetric finite element for the NASTRAN computer program has been presented. The element is an eight noded isoparametric quadrilateral ring with an assumed quadratic displacement and temperature function. The element has been named CQDRNG8. Element capabilities include elastic stiffness for homogeneous isotropic materials, thermoelastic loading, heat conduction for homogeneous isotropic materials, deformed and undeformed plotting, and stress recovery. Stress invariants calculated include the principal stresses and directions, the maximum shear, and the octahedral shear. The user has the option of specifying a 2 by 2, 3 by 3, or 4 by 4 order of Gaussian integration. To be consistent with other NASTRAN elements, the capability to specify arbitrary output coordinate systems has been included. Stiffness matrix generation times are comparable to other NASTRAN elements. On the IBM 370/158 series computer, stiffness matrix generation time is .30 seconds. Since the element is installed in the NASTRAN program, all of the general conveniences and capabilities of NASTRAN are also available. Some of these are convenient input, essentially unlimited problem size, efficient matrix operations, and a limited mesh generation capability. Grid locations can be generated in NASTRAN as usual but element connections cannot be generated. A simple FORTRAN program has been written by the author to generate element connection cards consistent with the generation scheme of the NASTRAN mesh generation program MSGMESH.

The accuracy and usefulness of the element has been demonstrated in several sample problems. The first problem was a commonly encountered bending problem--a simply supported circular plate. Deflections were within 2.5% and stresses were within 12.0% of theory. The second problem demonstrates the thermoelastic loading capability. An axially constrained flat circular plate with center hole was subjected to a uniform temperature change resulting in stress free radial and axial expansion of the plate. Calculated deflections were exactly those expected. The third case is a thermoelastic loading resulting in bending in a hollow sphere. Calculated bending stresses are within 8.0% of theory. The fourth problem demonstrates the heat conductance capability and element convergence. Internal temperatures are determined in a circular disk with center hole for several mesh patterns. Temperatures at both the inner and outer diameter are enforced. With only a single CQDRNG8 element, the predicted temperatures were within 12.0% of theory. Refinements reduced the error to as little as .25% and a maximum of 10.0%. The same problem was solved using another NASTRAN element, the CTRIARG. In all but the coarsest model, the CQDRNG8 predicts temperatures with a moderate increase in accuracy. For the coarse model, the results are the same. The fifth test problem is very similar to the fourth. A rod oriented axially has temperatures enforced at its ends. Calculated temperatures agree exactly with theory. The final problem provides a somewhat more severe test of the heat transfer capability. A hollow sphere has enforced temperatures at the inner diameter and convection at the outer diameter. Calculated results deviate from theory by less than 2.0%.

The first three sample problems clearly demonstrate the accuracy and

efficiency of the CQDRNG8 element for elastic and thermoelastic problems while the last three problems demonstrate the same qualities for heat transfer problems. Convergence is clearly shown. In all the sample problems, the grid mesh is relatively coarse. In the most severe tests (the plate bending and the hollow sphere bending and heat transfer), only two elements are present through the direction of largest gradient. The dissimilar nature of the problems illustrate the versatility of this element.

Some limitations do exist for this element. First the matrix generation is not as efficient as it could be. This is because of the technique available in NASTRAN at the time of the element's development. Matrix generation time could probably be reduced to less than one-fourth of the current value through use of a new technique available soon. The old technique forces repeated calculation of many quantities. The new technique eliminates this inefficiency. The matrix generation capability should be rewritten to incorporate this new technique when available.

The second limitation involves stresses. This element suffers from the same deficiencies as all displacement method finite elements. Interelement compatibility of stresses is not ensured. This leads to discontinuties of stresses at common grid points in adjacent elements. Realistic stress values at grid points are obtained by averaging the stresses in all common elements. The results presented in this paper were obtained in this manner. Some variation in individual grid point stresses existed but in all cases when these stresses were averaged, the results were quite close to theoretical stresses.

There are two aspects of this element which deserve further investigation. First a more complete convergence study should be undertaken and the convergence of this element compared to other existing axisymmetric finite elements. Secondly the effect of aspect ratio should be investigated. All of the sample problems were modeled with quite regular elements. Tests should include variations in length to width ratios, significant variations in included angles, and irregular spacing of midside grids. If the element is found to be highly effected by these variations, techniques for eliminating this dependence should be developed.

VIII. REFERENCES

1. Janucik, F.X., "Development and Applications of a Quadratic Isoparametric Finite Element for Axisymmetric Stress and Deflection Analysis", MS Thesis, Rochester Institute of Technology, Rochester, New York, (1974).

2. Zienkiewicz, O. C., The Finite Element Method in Engineering Science, (McGraw Hill Book Company, Inc., London, 1971).

3. Zienkiewicz, O. C. and Cheung, Y. K., The Finite Element Method in Structural and Continuum Mechanics (McGraw Hill Book Company, Inc., London, 1967).

4. Przemieniecki, J. S., Theory of Matrix Structural Analysis, (McGraw Hill Book Company, Inc., New York, 1968).

5. Segerlind, L. J., Applied Finite Element Analysis, (John Wiley and Sons, Inc., New York, 1976).

6. Desai, C. S. and Abel, J. F., Introduction to the Finite Element Method (Van Nostrand Reinhold Company, New York, 1972).

7. Bathe, K. J. and Wilson, E. L., Numerical Methods in Finite Element Analysis (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1976).

8. Ergatoudis, I., Irons, B. M., and Zienkiewicz, O. C., "Curved Isoparametric 'Quadrilateral' Elements for Finite Element Analysis", International Journal of Solids and Structures, Vol. 4, 31-42, (1968).

9. Baldwin, J. T., Razzaque, A., and Irons, B. M., "Shape Function Subroutine for an Isoparametric Thin Plate Element", Int. Journal Numerical Methods in Engineering, Vol. 7, 431-440, (1973).

10. Lee, H. P., NASTRAN Thermal Analyzer (NASA, Goddard Space Flight Center, Greenbelt, Maryland, 1975).

11. MacNeal, R. H., NASTRAN Theoretical Manual (The MacNeal-Schwendler Corporation, Los Angeles, 1974).

12. Doherty, W. P., Wilson, E. L., and Taylor, R. L., Stress Analysis of Axisymmetric Solids Utilizing Higher-Order Quadrilateral Finite Elements, (Structural Engineering Laboratory, University of California, Berkeley, 1969).

13. Bruch, J. C. and Zyroloski, G., "Transient Two-Dimensional Heat Conduction Problems Solved by the Finite Element Method", Int. J. Num. Meth. Eng., Vol. 8, 481-494, (1974).

14. Kohler, W. and Pittr, J., "Calculation of Transient Temperature Fields with Finite Elements in Space and Time Dimensions", Int. J. Num. Meth. Eng., Vol. 8, 625-632, (1974).

15. Zienkiewicz, O. C., and Parekh, C. J., "Transient Field Problems: Two-Dimensional and Three-Dimensional Analysis by Isoparametric Finite Elements", Int. J. Num. Meth. Eng., Vol. 2, 61-72, (1970).

16. Wall, S. E., MSC/NASTRAN Programmer's Manual,(The MacNeal-Schwendler Company, Los Angeles, 1976).

17. Wall, S. E., Programmer's Guide to MSC/NASTRAN, (The MacNeal-Schwendler Company, Los Angeles, 1974).

18. McCormick, C. W., MSC/NASTRAN User's Manual, (The MacNeal-Schwendler Company, Los Angeles, 1977).

19. McCormick, C. W., NASTRAN Basic Training Manual, (The MacNeal-Schwendler Company, Los Angeles, 1973).

20. Joseph, J. A., MSC/NASTRAN Applications Manual for IBM 360/370 Series, (The MacNeal-Schwendler Company, Los Angeles, 1976).

21. Timoshenko, S., and Krieger, S., Theory of Plates and Shells, (McGraw Hill Book Company, Inc., New York, 1959).

22. Carslaw, H. S. and Jaeger, J. C., Conduction of Heat in Solids, (Oxford University Press, London, 1959), second edition.

23. Schlottner, G., "Transient Thermal Stresses in Long Thick-Walled Cylinders", 1969.

24. Genberg, V., Eastman Kodak Company, Rochester, N.Y., Private Communication.

25. Mueller, P., Eastman Kodak Company, Rochester, N. Y., Private Communication.

26. Walter, W., Rochester Institute of Technology, Rochester, New York, Private Communication.

27. Wall, S., The MacNeal-Schwendler Company, Los Angeles, Private Communication.

28. Joseph, J., The MacNeal-Schwendler Company, Los Angeles, Private Communication.

29. MacNeal, R., The MacNeal-Schwendler Company, Los Angeles, Private Communication.

30. Rieger, N., Rochester Institute of Technology, Rochester, New York, Private Communication.

31. Harder, R., The MacNeal-Schwendler Company, Los Angeles, Private Communication.

IX. APPENDIX A

This section contains a listing of the subroutines needed for the new element capability, a functional description of the external subroutines used, and instructions for implementing the new subroutines. The functions of the various subroutines are as follows:

KDUM3	calculates stiffness or conductance matrix partitions
SMINIT	initializes Gaussian integration variables for subroutine KDUM3
SHAPE8	calculates shape functions and derivatives at the desired Gauss point for subroutine KDUM3
PDUM3	sets up connection arrays for plotting
DUM3	calculates equivalent nodal forces for thermal loading
SSGINT	performs same function as SMINIT for routine DUM3
SSHAP8	performs same function as SHAPE8 for routine DUM3
SDUM31	performs phase 1 stress calculations
SDUM32	performs phase 2 stress calculations
LTMP	calculates integration point temperature for temperature loading after linearizing the temperature distribution over the element; called by SDUM32
LINTMP	performs same function as LTMP for DUM3
MSHAP8	performs same function as SHAPE8 for SDUM32
ODUM3	formats and outputs stresses

Notice that all of the subroutines perform operations on the element level only--calculating the appropriate element dependent matrices.

These element matrices are calculated and added to the overall structural matrices. All structural matrix operations are performed using the existing NASTRAN techniques. The reader is referred to the manual by MacNeal for detailed descriptions of NASTRAN's many numerical techniques.

To incorporate the new element subroutines, the subroutines must be compiled and link-edited into the appropriate parts of NASTRAN. The specific details for accomplishing these tasks are computer and installation dependent. NASTRAN on IBM systems consists of 15 separate programs called LINK's. One link is a master link which controls the other links in such a way that NASTRAN appears to be a single program. The new subroutines must be link-edited into the specific links. The pertinent links and subroutines to be added to the links are:

LINK NAMES	SUBROUTINES TO BE ADDED TO LINK
LINKNS02	PDUM3
LINKNS03	KDUM3, SMINIT, SHAPE8
LINKNS05	DUM3, SSGINIT, SSHAPE8, LINTMP
LINKNS13	SDUM31, SDUM32, SD1INT, MSHAP8, LTMP
LINKNS14	ODUM3

The link-edit operations are performed using installation dependent JCL procedures and linkage editor control cards. Both of these are supplied by The MacNeal-Schwendler Company with each version of MSC/NASTRAN.

EXTERNAL SUBROUTINES

NAME	SOURCE	DESCRIPTION
MESAGE	NASTRAN utility	Prints NASTRAN error messages
MAT	NASTRAN utility	Obtains elastic material properties
HMAT1	NASTRAN utility	Obtains thermal material properties
GMMATD	NASTRAN utility	General matrix multiplication
INVERD	NASTRAN utility	Incore matrix inversion
TRANSD	NASTRAN utility	Obtains coordinate transformation matrices
SMA1B	NASTRAN utility	Matrix insertion routine
FREAD	NASTRAN utility	Read NASTRAN table
SSGETD	NASTRAN utility	Obtains element temperature data
DABS,ABS	FORTRAN library	Absolute value
BASGLB	NASTRAN utility	Converts vector from basic to global coordinates
SDR2VR	NASTRAN utility	Obtains displacements
PRNSTR	NASTRAN utility	Computes 3-D principal stresses
SQRT	FORTRAN library	Computes square root

```
      SUBROUTINE KDUM3
C  *****
C THIS ROUTINE CALCULATES THE EIGHT 6X6 MATRICES K(NPVT,J) FOR
C A QDRNG8 ELEMENT HAVING CONNECTIONS NPVT AND J=1 THRU 8
C   REQUIRES  ADUM3,8,,,1,3,CQDRNG8   DATA CARD
C *****
C
C          E C P T     F O R     T H E     Q D R N G 8
C                                                 TYPE TABLE CARD
C
C ECPT( 1)  ELEMENT ID                              I    ECT  CDUM
C ECPT( 2)  SCALAR INDEX NUMBER FOR GRID A          I    ECT  CDUM
C ECPT( 3)  SCALAR INDEX NUMBER FOR GRID B          I    ECT  CDUM
C ECPT( 4)  SCALAR INDEX NUMBER FOR GRID C          I    ECT  CDUM
C ECPT( 5)  SCALAR INDEX NUMBER FOR GRID D          I    ECT  CDUM
C ECPT( 6)  SCALAR INDEX NUMBER FOR GRID E          I    ECT  CDUM
C ECPT( 7)  SCALAR INDEX NUMBER FOR GRID F          I    ECT  CDUM
C ECPT( 8)  SCALAR INDEX NUMBER FOR GRID G          I    ECT  CDUM
C ECPT( 9)  SCALAR INDEX NUMBER FOR GRID H          I    ECT  CDUM
C ECPT(10)  MATERIAL ID                             I    EPT  PDUM
C ECPT(11)  NO OF INTEGRATION GAUSS POINTS          I    EPT  PDUM
C ECPT(12)  DISP COORD SYS ID FOR GRID A            I  BGPDT GRID
C ECPT(13)  X COORD OF GRID A (BASIC)               R  BGPDT GRID
C ECPT(14)  Y COORD OF GRID A (BASIC)               R  BGPDT GRID
C ECPT(15)  Z COORD OF GRID A (BASIC)               R  BGPDT GRID
C ECPT(16)  DISP COORD SYS ID FOR GRID B            I  BGPDT GRID
C ECPT(17)  X COORD OF GRID B (BASIC)               R  BGPDT GRID
C ECPT(18)  Y COORD OF GRID B (BASIC)               R  BGPDT GRID
C ECPT(19)  Z COORD OF GRID B (BASIC)               R  BGPDT GRID
C ECPT(20)  DISP COORD SYS ID FOR GRID C            I  BGPDT GRID
C ECPT(21)  X COORD OF GRID C (BASIC)               R  BGPDT GRID
C ECPT(22)  Y COORD OF GRID C (BASIC)               R  BGPDT GRID
C ECPT(23)  Z COORD OF GRID C (BASIC)               R  BGPDT GRID
C ECPT(24)  DISP COORD SYS ID FOR GRID D            I  BGPDT GRID
C ECPT(25)  X COORD OF GRID D (BASIC)               R  BGPDT GRID
C ECPT(26)  Y COORD OF GRID D (BASIC)               R  BGPDT GRID
C ECPT(27)  Z COORD OF GRID D (BASIC)               R  BGPDT GRID
C ECPT(28)  DISP COORD SYS ID FOR GRID E            I  BGPDT GRID
C ECPT(29)  X COORD OF GRID E (BASIC)               R  BGPDT GRID
C ECPT(30)  Y COORD OF GRID E (BASIC)               R  BGPDT GRID
C ECPT(31)  Z COORD OF GRID E (BASIC)               R  BGPDT GRID
C ECPT(32)  DISP COORD SYS ID FOR GRID F            I  BGPDT GRID
C ECPT(33)  X COORD OF GRID F (BASIC)               R  BGPDT GRID
C ECPT(34)  Y COORD OF GRID F (BASIC)               R  BGPDT GRID
C ECPT(35)  Z COORD OF GRID F (BASIC)               R  BGPDT GRID
C ECPT(36)  DISP COORD SYS ID FOR GRID G            I  BGPDT GRID
C ECPT(37)  X COORD OF GRID G (BASIC)               R  BGPDT GRID
C ECPT(38)  Y COORD OF GRID G (BASIC)               R  BGPDT GRID
C ECPT(39)  Z COORD OF GRID G (BASIC)               R  BGPDT GRID
C ECPT(40)  DISP COORD SYS ID FOR GRID H            I  BGPDT GRID
C ECPT(41)  X COORD OF GRID H (BASIC)               R  BGPDT GRID
C ECPT(42)  Y COORD OF GRID H (BASIC)               R  BGPDT GRID
```

```
C ECPT(42)  Z COORD OF GRID H (BASIC)          R    PGPDT GRID
C ECPT(44)  ELEMENT TEMPERATURE
C
C
C
      DOUBLE PRECISION KE,TI,TJ,INDEX,VOL
      DOUBLE PRECISION K,RZ,N,DN,DNDRZ,JAC,PP,QQ,W,P,Q,RAD,DETERM,
     1   PI,E,NU,DE,DBAR,G,DUM
      REAL NUS
      LOGICAL HEAT
      DIMENSION IECPT(1)
      COMMON /SMA1DP/ K(32),RZ(16),N(8),DN(16),DNDRZ(16),JAC(2,2),
     1   PP(16),QQ(16),W(16),P(8),Q(8),RAD,DETERM,E,NU,DE,DBAR,G,
     2   KE(36),TI(9),TJ(9),INDEX(2,3),VOL,DUM(76)
      COMMON /CONDAD/ PI,TWOPI,RADDEG,DEGRAD,FORPI2
      COMMON /MATIN/  MATID,MATFLG,ELTEMP,STRESS,SINTH,COSTH
      COMMON /MATOUT/ ES,GS,NUS,RHO,ALPHA,TSUBO,GSUBE,SIGT,SIGC,SIGS
      COMMON /HMTOUT/ EK
      COMMON /SMA1ET/ ECPT(100)
      COMMON /SMA1ID/ SKP(10),IFKGG,SKP1,IF4GG,SKP3(23)
      COMMON /SMA1CL/ IOPT,K4GGSW,NPVT,SKP2(19),NOGO
      COMMON /SMA1HT/ HEAT
      EQUIVALENCE (  ECPT(1),IECPT(1)  )
C        CHECK FOR NPVT GRID
      DO 5 I=1,8
      IF(NPVT.NE.IECPT(I+1) ) GO TO  5
      NONPVT = I
      GO TO 8
    5 CONTINUE
C        ERROR ** NO PIVOT GRID FOUND
      CALL MESAGE(-30,34,IECPT(1))
      NOGO=1
    8 CONTINUE
C        FILL  MATIN  COMMON  BLOCK
      MATID = IECPT(10)
      MATFLG = 1
      ELTEMP = IECPT(44)
C        CHECK FOR HEAT FORMULATION
      IF(HEAT)GO TO 9
C     I  RETRIEVE STRUCTURAL MATERIAL PROPERTIES
      CALL MAT(IECPT(1) )
C        STORE  PROPERTIES IN DOUBLE PRECISION LOCATIONS
      E = ES
      NU= NUS
C        CHECK  FOR  ILLEGAL  PROPERTIES
      IF( (E.EQ.0.0D0).OR.(NU.EQ.0.0D0) ) GO TO 31
C        CHECK FOR POSSIBLE DIVIDE BY ZERO
      IF( NU.GE.0.5D0 ) GO TO 30
      DE = NU / (1.D0 - NU )
      G  = (1.D0 - 2.D0 * NU )/(2.D0 * (1.D0 - NU ) )
      DBAR= E * (1.D0 - NU)/( (1.D0 + NU) * (1.D0 - 2.D0 * NU ) )
      DBAR = DBAR * TWOPI
```

```
      GO TO 35
C        RETRIEVE  HEAT  PROPERTIES
    9 CALL HMAT1( IECPT(1)  )
      E= EK
      IF( E.EQ.0.DO ) GO TO 30
      E = E * TWOPI
      GO TO 35
C        NU IS GE .5
   30 CALL MESAGE(-30,16,IECPT(1).)
      CALL MESAGE(-30,40,IECPT(1))
      NOGO = 1
      GO TO 35
C        E = 0. OR  NU = 0.
   31 CALL MESAGE(-30,126,IECPT(1))
      NOGO = 1
      GO TO 35
C        CHECK  NOGO  FLAG
   35 IF(NOGO.EQ.1) GO TO 5000
C        WRITE  R AND Z  COORINATES  INTO  DOUBLE  PRECISION  ARRAY  RZ
      DO 40  I=1,8
      JJ = 1 + 2*(I-1)
      J  = 13 + 4*(I-1)
      RZ(JJ  ) = ECPT(J)
      RZ(JJ+1) = ECPT(J+2)
   40 CONTINUE
C        ZERO  K  MATRIX
      DO  50  I=1,32
      K(I) = 0.DO
   50 CONTINUE
C        INITIALIZE  VARIABLES  FOR  GAUSSIAN  INTEGRATION
      NGP = IECPT(11)
      CALL  SMINIT(NGP)
C        TOP  OF  INTEGRATION  LOOP
      DO  900  INT =1,NGP
C        CALCULATE  SHAPE  FUNCTIONS  AND  LOCAL  DERIVATIVES
      CALL  SHAPE8( INT )
C        CALCULATE  JACOBIAN     J = DN * RZ    STORE  IN  DUM
      CALL  GMMATD(DN,2,8,0,RZ,8,2,0,DUM)
C        STORE  DUM  IN  DOUBLY  DIMENSIONED  ARRAY  JAC(2,2)
      JAC(1,1) = DUM(1)
      JAC(1,2) = DUM(2)
      JAC(2,1) = DUM(3)
      JAC(2,2) = DUM(4)
C        CALCULATES  INVERSE  JACOBIAN
C        RETURNS  INVERSE ,  DETERMINANT ,  SINGULARITY
      CALL  INVERD( 2,JAC,2,DUM,0,DETERM,ISING,INDEX )
C        CHECK  FOR  SINGULARITY
      IF( ISING.EQ.2 ) GO TO 400
C        CHECK  FOR  ZERO  DETERMINANT
      IF( DETERM.EQ.0.DO ) GO TO 500
C        STORE  INVERSE  JACOBIAN  IN  VECTOR  DUM
      DUM(1)  =  JAC(1,1)
```

```
      DUM(2)  =  JAC(1,2)
      DUM(3)  =  JAC(2,1)
      DUM(4)  =  JAC(2,2)
C        CALCULATE  DNDRZ = INVERSE JAC * DN
      CALL GMMATD(DUM,2,2,0,DN,2,8,0,DNDRZ)
C        CALCULATE  RADIUS
      RAD = 0.DO
      DO  60  I=1,8
      J = 1 + 2*(I-1)
      RAD = RAD + N(I)*RZ(J)
   60 CONTINUE
C        CALCULATE  VOLUME  FOR  THIS  GAUSS  POINT
      VOL = RAD * W(INT) * DABS(DETERM)
      IF( RAD.NE.0.DO ) GO TO 80
C        ZERO  RADIUS
      CALL  MESAGE(-30,32,IECPT(1) )
      NOGO = 1
      RETURN
C        SINGULAR  JACOBIAN
  400 CALL MESAGE(-30,33,IECPT(1) )
      NOGO = 1
      IF( DETERM.NE.0.DO ) GO TO 5000
C        DETERMINANT  OF  JACOBIAN  =  0
  500 CALL  MESAGE(-30,37,IECPT(1) )
      NOGO = 1
      GO TO 5000
   80 CONTINUE
C        CALCULATE EIGHT 2 BY 2 STIFFNESS PARTITIONS AND STORE EACH AS A
C        VECTOR       K(1)=KELM(1,1)
C                     K(2)=KELM(1,2)
C                     K(3)=KELM(2,1)
C                     K(4)=KELM(2,2)
C        K( 1)  THRU  K( 4)  IS  FIRST    2 BY 2  PARTITION
C        K( 5)  THRU  K( 8)  IS  SECOND   2 BY 2  PARTITION
C        K( 9)  THRU  K(12)  IS  THIRD    2 BY 2  PARTITION
C        K(13)  THRU  K(16)  IS  FOURTH   2 BY 2  PARTITION
C        K(17)  THRU  K(20)  IS  FIFTH    2 BY 2  PARTITION
C        K(21)  THRU  K(24)  IS  SIXTH    2 BY 2  PARTITION
C        K(25)  THRU  K(28)  IS  SEVENTH  2 BY 2  PARTITION
C        K(29)  THRU  K(32)  IS  EIGHTH   2 BY 2  PARTITION
      IZ = 8 + NONPVT
      IF(HEAT) GO TO 700
      DO 600  JR = 1,8
      JZ=8 + JR
      JJ=1 + 4*(JR-1)
      K(JJ)=VOL*( DNDRZ(NONPVT)*( DNDRZ(JR)+DE*N(JR)/RAD )
     1 + ( N(NONPVT)/RAD )*( DE*DNDRZ(JR) + N(JR)/RAD )
     2 + ( DNDRZ(IZ)*DNDRZ(JZ)*G )  )   +  K(JJ)
      K(JJ+1) = VOL*( DNDRZ(NONPVT)*DNDRZ(JZ)*DE + N(NONPVT)*DNDRZ(JZ)
     1          *DE/RAD + DNDRZ(IZ)*DNDRZ(JR)*G )        + K(JJ+1)
      K(JJ+2) = VOL*( DNDRZ(IZ)*DE* ( DNDRZ(JR)     + N(JR)/RAD )
     1          + DNDRZ(NONPVT)*DNDRZ(JZ)*G )   + K(JJ+2)
```

```
      K(JJ+3) = VOL*( DNDRZ(IZ)*DNDRZ(JZ) + DNDRZ(NONPVT)*DNDRZ(JR)*G )
     1          +  K(JJ+3)
  600 CONTINUE
      GO TO 900
C        HEAT  FORMULATION
  700 DO 750 JR=1,8
      JZ = 8 + JR
      K(JR  ) = VOL*( DNDRZ(NONPVT)*DNDRZ(JR) + DNDRZ(IZ)*DNDRZ(JZ) )
     1          +   K(JR)
  750 CONTINUE
  900 CONTINUE
C        BOTTOM  OF  INTEGRATION  LOOP
      IF(HEAT) GO TO 1200
C        RETRIEVE  PIVOT  GRID  TRANSFORMATION  SYSTEM
      KA = 12 +  4*(NONPVT -1)
      CALL  TRANSD( IECPT(KA),TI )
C        EXPAND 2 BY 2 PARTITIONS TO 3 BY 3 AND MULTIPLY BY CONSTANT
      DO  1000  J=1,8
C        ZERO KE MATRIX
      DO  910   I=1,36
      KE(I)=0.DO
  910 CONTINUE
      KB = 12 + 4*(J-1)
      JJ = 1  + 4*(J-1)
      KE(1) =  K(JJ  ) * DBAR
      KE(2) =  K(JJ+1) * DBAR
      KE(7) =  K(JJ+2) * DBAR
      KE(9) =  K(JJ+3) * DBAR
C        CHECK  IF  TRANSFORMATION  IS  NEEDED
      IF( (IECPT(KA).EQ.0).AND.(IECPT(KB).EQ.0) ) GO TO 990
C        FORM  TIT  *  KE
      CALL  GMMATD(TI,3,3,1,KE,3,3,0,RZ )
C        RETRIEVE TJ
      CALL  TRANSD( IECPT(KB),TJ )
C        FORM  TIT  *  KE  *  TJ
      CALL  GMMATD( RZ,3,3,0,TJ,3,3,0,KE )
C        EXPAND  3 BY 3  TO  6 BY 6
  990 KE(15) = KE( 9)
      KE(14) = KE( 8)
      KE(13) = KE( 7)
      KE( 9) = KE( 6)
      KE( 8) = KE( 5)
      KE( 7) = KE( 4)
      KE( 6) = 0.DO
      KE( 5) = 0.DO
      KE( 4) = 0.DO
C        CALL  INSERTION  ROUTINE
      CALL  SMA1B( KE,IECPT(J+1),0,IFKGG,0.DO )
 1000 CONTINUE
      RETURN
C        HEAT  FORMULATION  INSERTION
 1200 DO  1250  I=1,8
```

```
      DE = F *  K(I)
      CALL  SMA1B( DE,IECPT(I+1),NPVT,IFKGG,0.D0 )
1250  CONTINUE
5000  RETURN
      END
```

```
      SUBROUTINE SMINIT(NGP)
C        INITIALIZES  GAUSSIAN  INTEGRATION  VARIABLES
C        ARRAYS  P AND Q  CONTAIN  LOCAL  GRID  COORDINATES
C        ARRAYS  PP, QQ, AND W  CONTAIN  GAUSS  POINT  COORDINATES  AND
C        WEIGHTING  FUNCTIONS
      DOUBLE PRECISION KE,TI,TJ,INDEX,VOL
      DOUBLE  PRECISION  K,RZ,N,DN,DNDRZ,JAC,PP,QQ,W,P,Q,RAD,DETERM,
     1    PI,E,NU,DE,DBAR,G,DUM
      COMMON  /SMA1DP/ K(32),RZ(16),N(8),DN(16),DNDRZ(16),JAC(2,2),
     1   PP(16),QQ(16),W(16),P(8),Q(8),RAD,DETERM,E,NU,DE,DBAR,G,
     2   KE(36),TI(9),TJ(9),INDEX(2,3),VOL,DUM(76)
      P(1)=-1.D0
      Q(1)=-1.D0
      P(2)= 0.D0
      Q(2)=-1.D0
      P(3)=+1.D0
      Q(3)=-1.D0
      P(4)=+1.D0
      Q(4)= 0.D0
      P(5)=+1.D0
      Q(5)=+1.D0
      P(6)= 0.D0
      Q(6)= 1.D0
      P(7)=-1.D0
      Q(7)=+1.D0
      P(8)=-1.D0
      Q(8)= 0.D0
C        DETERMINE  NUMBER  OF  GAUSS  POINTS  TO  BE  USED
      GO TO ( 300,200,300,400 ) , NGP
C        3 BY 3  GAUSSIAN  INTEGRATION
  300 PP(1) =-.774596669241483D0
      QQ(1) =PP(1)
      DUM(1)=.555555555555556D0
      DUM(2)= .888888888888889D0
      W(1)  = DUM(1) * DUM(1)
      PP(2) = 0.D0
      QQ(2) = QQ(1)
      W(2)  = DUM(1) * DUM(2)
      PP(3) =- PP(1)
      QQ(3) =  QQ(1)
      W(3)  = W(1)
      PP(4) =  PP(1)
      QQ(4) = 0.D0
      W(4)  = W(2)
      PP(5) = 0.D0
      QQ(5) = 0.D0
      W(5)  = DUM(2) * DUM(2)
      PP(6) =  PP(3)
      QQ(6) =  0.D0
      W(6)  =  W(2)
      PP(7) =  PP(1)
      QQ(7) = -QQ(1)
```

```
      W(7)   =  W(1)
      PP(8)  =  0.D0
      QQ(8)  =  QQ(7)
      W(8)   =  W(2)
      PP(9)  =  PP(3)
      QQ(9)  =  QQ(7)
      W(9)   =  W(1)
      NGP    =  9
      RETURN
C          2 BY 2  GAUSSIAN  INTEGRATION
  200 PP(1)=-.577350269189626D0
      QQ(1)=  PP(1)
      PP(2)= -PP(1)
      QQ(2)=  QQ(1)
      PP(3)=  PP(1)
      QQ(3)= -QQ(1)
      PP(4)=  PP(2)
      QQ(4)=  QQ(3)
      DO 250 I=1,4
      W(I)= 1.D0
  250 CONTINUE
      NGP = 4
      RETURN
C          4 BY 4  GAUSSIAN  INTEGRATION
  400 DUM(1) = .652145154862546D0
      DUM(2) = .347854845137454D0
      PP(1)  =-.861136311594053D0
      QQ(1)  = PP(1)
      W(1)   = DUM(2) * DUM(2)
      PP(2)  =-.339981043584856D0
      QQ(2)  = QQ(1)
      W(2)   = DUM(1) * DUM(2)
      PP(3)  =-PP(2)
      QQ(3)  = QQ(1)
      W(3)   = W(2)
      PP(4)  =-PP(1)
      QQ(4)  = QQ(1)
      W(4)   = W(1)
      PP(5)  = PP(1)
      QQ(5)  = PP(2)
      W(5)   = W(2)
      PP(6)  = PP(2)
      QQ(6)  = QQ(5)
      W(6)   = DUM(1) * DUM(1)
      PP(7)  = PP(3)
      QQ(7)  = QQ(5)
      W(7)   = W(6)
      PP(8)  = PP(4)
      QQ(8)  = QQ(5)
      W(8)   = W(2)
      PP(9)  = PP(1)
      QQ(9)  =-QQ(5)
```

```
W(9)    = W(2)
PP(10)  = PP(2)
QQ(10)  = QQ(9)
W(10)   = W(6)
PP(11)  = PP(3)
QQ(11)  = QQ(9)
W(11)   = W(6)
PP(12)  = PP(4)
QQ(12)  = QQ(9)
W(12)   = W(2)
PP(13)  = PP(1)
QQ(13)  =-QQ(1)
W(13)   = W(1)
PP(14)  = PP(2)
QQ(14)  = QQ(13)
W(14)   = W(2)
PP(15)  = PP(3)
QQ(15)  = QQ(13)
W(15)   = W(2)
PP(16)  = PP(4)
QQ(16)  = QQ(13)
W(16)   = W(1)
NGP     =  16
RETURN
END
```

```
      SUBROUTINE  SHAPE8(INT)
C         CALCULATES  N(I)  AND  LOCAL  DERIVATIVES  DN(I)/DP , DN(I)/DQ
C         FOR  EIGHT  NODED  ISOPARAMETRIC  QUADRILATERAL  RING  ELEMENT
      DOUBLE PRECISION KE,TI,TJ,INDEX,VOL
      DOUBLE  PRECISION  K,KZ,N,DN,DNDRZ,JAC,PP,QQ,W,P,Q,RAD,DETERM,
     1    PI,E,NU,DE,DEAR,G,DUM
      DOUBLE PRECISION PO,QO
      COMMON  /SMA1DP/ K(32),KZ(16),N(8),DN(16),DNDRZ(16),JAC(2,2),
     1    PP(16),QQ(16),W(16),P(8),Q(8),RAD,DETERM,E,NU,DE,DEAR,G,
     2    KE(36),TI(9),TJ(9),INDEX(3,3),VOL,DUM(76)
C         PP  AND  QQ  CONTAIN  COORDINATES  OF  GAUSS  INTEGRATION  POINTS
C         INT IS  THE  INDEX  OF  CURRENT  GAUSS  POINT  BEING  CONSIDERED
C         ODD  INDICES     /     CORNER NODES
      DO 10 I=1,7,2
      PO = PP(INT) * P(I)
      QO = QQ(INT) * Q(I)
C         CALCULATE  SHAPE  FUNCTIONS  N
      N(I) = .25D0*(1.D0+PO)*(1.D0+QO)*(PO+QO-1.D0)
C         CALCULATE  DERIVATIVE  WRT  LOCAL  COORDINATE  P  -  DN(I)/DP
      DN(I) =.25D0*P(I)*(1.D0+QO)*(2.D0*PO + QO)
C         CALCULATE  DERIVATIVE  WRT  LOCAL  COORDINATE  Q  -  DN(I)/DQ
      DN(I+8)=.25D0*Q(I)*(1.D0+PO)*(PO+2.D0*QO)
   10 CONTINUE
C         EVEN  INDICES     /     MIDSIDE NODES
      DO 20 I=2,8,2
      PO = PP(INT) * P(I)
      QO = QQ(INT) * Q(I)
      N(I) = .5D0 * ( 1.D0-PP(INT)*PP(INT) ) * ( 1.D0 + QO ) *Q(I)*Q(I)
     1 + .5D0 * ( 1.D0-QQ(INT)*QQ(INT) ) * ( 1.D0 + PO ) *P(I) * P(I)
      DN(I) = -PP(INT) * ( 1.D0 + QO ) * Q(I) * Q(I)
     1    + .5D0 * ( 1.D0 - QQ(INT)*QQ(INT) ) * P(I)
      DN(I+8) = -QQ(INT) * ( 1.D0 + PO ) * P(I) * P(I)
     1    +.5D0 * ( 1.D0 - PP(INT)*PP(INT) ) * Q(I)
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  PDUM3 (*,*,*,IZ,M,NZ,NGPEL,K,IELS,IOPT)
C  SETS UP CONNECTION ARRAYS FOR PLOTTING
      DIMENSION  IZ(1),IELS(1),IGLS(8)
      M = 10
      IF( IOPT.EQ.0 ) M = 16
      IF( NZ.LT.M ) RETURN 3
    5 CALL FREAD( IELS,IEID,1,0 )
      IF( IEID.EQ.0 ) RETURN 1
      CALL FREAD( IELS,IGLS,NGPEL,0 )
      IF( IOPT.NE.0 ) GO TO 50
      DO 10 I=1,7
      IZ(K) = IGLS(I)
      IZ(K+1)= IGLS(I+1)
      K=K+2
   10 CONTINUE
      IZ(K  ) = IGLS(8)
      IZ(K+1) = IGLS(1)
      K=K+2
      NZ= NZ - M
      IF( NZ.LT.M ) RETURN 2
      GO TO 5
   50 DO 60 I=1,8
      IZ(K+I-1) = IGLS(I)
   60 CONTINUE
      IZ(K+8) = IGLS(1)
      IZ(K+9) = 0
      K=K+M
      NZ = NZ - M
      IF( NZ.LT.M ) RETURN  2
      GO TO 5
  100 RETURN
      END
```

```
      SUBROUTINE DUM3( PG )
C   CALCULATES THERMAL LOADS FOR THE CQDRNGP ELEMENT
      DOUBLE PRECISION TWOPI,RADDEG,DEGRAD,FORPI2
      DOUBLE PRECISION  P,Q,PP,QQ,W,N,DN
      DOUBLE PRECISION  FORCE
      DOUBLE PRECISION INDEX,DNDRZ,VOL,JAC,RAD,DETERM,RZ,DUM(4)
      DOUBLE PRECISION  PI
      DOUBLE PRECISION  ALPHA,E,NU,TSUBO,DE,DBAR,TBAR
      DOUBLE PRECISION G,D
      REAL  FORBAS(3) , FORGLB(3)
      REAL  PG(1)
      REAL  T(9)
      REAL NUS
      DIMENSION IECPT(1)
      COMMON  /SSGETT/  LTYPE,SKP1(4),ITEMP,IDEFT,IDEFM
      COMMON /MATIN/  MATID,MATFLG,ELTEMP,STRESS,SINTH,COSTH
      COMMON /MATOUT/ ES,GS,NUS,RHO,ALPHAS,TSUBOS,GSUBE,SIGT,SIGC,SIGS
      COMMON /TRIMEX/  ECPT(100)
      COMMON /SSGWRK/ P(8),Q(8),PP(16),QQ(16),W(16),N(8),DN(16),TBAR,
     1    FORCE(16),DNDRZ(16),JAC(2,2),INDEX(2,3),RZ(16),VOL,
     2    RAD,DETERM
      COMMON /CONDAD/ PI,TWOPI,RADDEG,DEGRAD,FORPI2
      EQUIVALENCE   (ECPT(1),IECPT(1)  )
      IF( ITEMP.LE.0 ) GO TO 1000
C        FILL  MATIN  COMMON  BLOCK
      MATID = IECPT(10)
      MATFLG = 1
      ELTEMP = IECPT(44)
C     I  RETRIEVE STRUCTURAL MATERIAL PROPERTIES
      CALL MAT(IECPT(1) )
C        STORE  PROPERTIES IN DOUBLE PRECISION LOCATIONS
      E = ES
      NU= NUS
      ALPHA = ALPHAS
      TSUBO = TSUBOS
C        CHECK  FOR  ILLEGAL  PROPERTIES
      IF( (E.EQ.0.D0).OR.(NU.EQ.0.D0) ) GO TO 31
C        CHECK FOR POSSIBLE DIVIDE BY ZERO
      IF( NU.GE.0.5D0 ) GO TO 30
      DE = NU / (1.D0 - NU )
      G  = (1.D0 - 2.D0 * NU )/(2.D0 * (1.D0 - NU ) )
      DBAR= E * (1.D0 - NU)/( (1.D0 + NU) * (1.D0 - 2.D0 * NU ) )
      DBAR = DBAR * TWOPI
      GO TO 35
C        NU IS GE .5
   30 CALL MESAGE(-30,16,IECPT(1) )
      CALL MESAGE(-30,40,IECPT(1))
      NOGO = 1
      GO TO 35
C        E = 0. OR  NU = 0.
   31 CALL MESAGE(-30,126,IECPT(1))
      NOGO = 1
```

```
      GO TO 35
C        CHECK  NOGO  FLAG
   35 IF(NOGO.EQ.1) GO TO 5000
C        WRITE  R AND Z  COORINATES  INTO  DOUBLE  PRECISION  ARRAY  RZ
      DO 40  I=1,8
      JJ = 1 + 2*(I-1)
      J  = 13 + 4*(I-1)
      RZ(JJ  ) = ECPT(J)
      RZ(JJ+1) = ECPT(J+2)
   40 CONTINUE
C  RETRIEVE  TEMPERATURE  DATA
      ID = IECPT(1)
      CALL SSGETD(ID,T,8)
C        INITIALIZE  VARIABLES  FOR  GAUSSIAN  INTEGRATION
      NGP = IECPT(11)
      CALL SSGINT( NGP )
C  ZERO  FORCE VECTOR
      DO 50  I=1,16
      FORCE(I) = 0.DO
   50 CONTINUE
C        TOP  OF  INTEGRATION  LOOP
      DO  900  INT =1,NGP
C        CALCULATE  SHAPE  FUNCTIONS  AND  LOCAL  DERIVATIVES
      CALL  SSHAP8 ( INT )
C        CALCULATE  JACOBIAN     J = DN * RZ    STORE  IN  DUM
      CALL  GMMATD(DN,2,8,0,RZ,8,2,0,DUM)
C        STORE  DUM  IN  DOUBLY  DIMENSIONED  ARRAY  JAC(2,2)
      JAC(1,1) = DUM(1)
      JAC(1,2) = DUM(2)
      JAC(2,1) = DUM(3)
      JAC(2,2) = DUM(4)
C        CALCULATES  INVERSE  JACOBIAN
C        RETURNS  INVERSE ,  DETERMINANT ,  SINGULARITY
      CALL  INVERD( 2,JAC,2,DUM,0,DETERM,ISING,INDEX )
C        CHECK  FOR  SINGULARITY
      IF( ISING.EQ.2 ) GO TO 400
C        CHECK  FOR  ZERO  DETERMINANT
      IF( DETERM.EQ.0.DO ) GO TO 500
C        STORE  INVERSE  JACOBIAN  IN  VECTOR  DUM
      DUM(1)  =  JAC(1,1)
      DUM(2)  =  JAC(1,2)
      DUM(3)  =  JAC(2,1)
      DUM(4)  =  JAC(2,2)
C        CALCULATE  DNDRZ = INVERSE JAC * DN
      CALL GMMATD(DUM,2,2,0,DN,2,8,0,DNDRZ)
C        CALCULATE  RADIUS
      RAD = 0.DO
      DO  60  I=1,8
      J = 1 + 2*(I-1)
      RAD = RAD + N(I)*RZ(J)
   60 CONTINUE
C        CALCULATE  VOLUME  FOR  THIS  GAUSS  POINT
```

```
      VOL = RAD * W(INT) * DABS(DETERM)
      IF( RAD.NE.0.DO ) GO TO 80
C        ZERO  RADIUS
      CALL  MESAGE(-30,32,IECPT(1) )
      NOGO = 1
      RETURN
C        SINGULAR  JACOBIAN
  400 CALL MESAGE(-30,33,IECPT(1) )
      NOGO = 1
      IF( DETERM.NE.0.DO ) GO TO 5000
C        DETERMINANT  OF  JACOBIAN  =  0
  500 CALL  MESAGE(-30,37,IECPT(1) )
      NOGO = 1
      GO TO 5000
   80 CONTINUE
      CALL  LINTMP(  INT,T  )
      TBAR = TBAR - TSUBO
      D = ( 1.DO + 2.DO * DE ) * ALPHA
      J=0
      DO  600  I=1,16,2
      J=J+1
      FORCE(I)=FORCE(I) +(DNDRZ(J)  + N(J)/ RAD )*TBAR * VOL
  600 CONTINUE
      J = 8
      DO  700  I=2,16,2
      J=J+1
      FORCE(I)=FORCE(I) +(DNDRZ(J) * TBAR * VOL  )
  700 CONTINUE
  900 CONTINUE
C        BOTTOM  OF  INTEGRATION  LOOP
C  CONVERT FROM BASIC  TO  GLOBAL  COORDINATES  AND  ADD TO  PG
      K=2
      J=12
      DO  910  I=1,16,2
      FORBAS(1)= FORCE(I)*D*DBAR
      FORBAS(2) = 0.0
      FORBAS(3)  =FORCE(I+1)*D*DBAR
      CALL  BASGLB (FORBAS,FORGLB,ECPT(J+1),IECPT(J)  )
      J = J +4
      PG( IECPT(K) )= FORGLB(1)  + PG( IECPT(K) )
      PG( IECPT(K)+1) =FORGLB(2) + PG( IECPT(K)+1 )
      PG( IECPT(K)+2) =FORGLB(3) + PG( IECPT(K)+2 )
      K=K+1
  910 CONTINUE
 1000 RETURN
 5000 RETURN
      END
```

```
      SUBROUTINE SSGINT( NGP )
C        INITIALIZES  GAUSSIAN  INTEGRATION  VARIABLES
C        ARRAYS  P AND Q  CONTAIN  LOCAL  GRID  COORDINATES
C        ARRAYS  PP, QQ, AND W  CONTAIN  GAUSS  POINT  COORDINATES  AND
C        WEIGHTING  FUNCTIONS
      DOUBLE PRECISION  P,Q,PP,QQ,W,N,DN
      DOUBLE PRECISION TBAR
      DOUBLE PRECISION DUM
      COMMON /SSGWRK/ P(8),Q(8),PP(16),QQ(16),W(16),N(8),DN(16),TBAR,
     1    DUM(61)
      P(1)=-1.D0
      Q(1)=-1.D0
      P(2)= 0.D0
      Q(2)=-1.D0
      P(3)=+1.D0
      Q(3)=-1.D0
      P(4)=+1.D0
      Q(4)= 0.D0
      P(5)=+1.D0
      Q(5)=+1.D0
      P(6)= 0.D0
      Q(6)= 1.D0
      P(7)=-1.D0
      Q(7)=+1.D0
      P(8)=-1.D0
      Q(8)= 0.D0
C        DETERMINE  NUMBER  OF  GAUSS  POINTS  TO  BE  USED
      GO TO ( 300,200,300,400 ) , NGP
C        3 BY 3  GAUSSIAN  INTEGRATION
  300 PP(1) =-.774596669241483D0
      QQ(1) =PP(1)
      DUM(1)=.555555555555556D0
      DUM(2)= .888888888888889D0
      W(1)  = DUM(1) * DUM(1)
      PP(2) = 0.D0
      QQ(2) = QQ(1)
      W(2)  = DUM(1) * DUM(2)
      PP(3) =- PP(1)
      QQ(3) =  QQ(1)
      W(3)  = W(1)
      PP(4) =  PP(1)
      QQ(4) = 0.D0
      W(4)  = W(2)
      PP(5) = 0.D0
      QQ(5) = 0.D0
      W(5)  = DUM(2) * DUM(2)
      PP(6) =  PP(3)
      QQ(6) =  0.D0
      W(6)  =  W(2)
      PP(7) =  PP(1)
      QQ(7) = -QQ(1)
      W(7)  =  W(1)
```

```
      PP(8) =  0.D0
      QQ(8) =  QQ(7)
      W(8)  =  W(2)
      PP(9) =  PP(3)
      QQ(9) =  QQ(7)
      W(9)  =  W(1)
      NGP   =  9
      RETURN
C        2 BY 2  GAUSSIAN  INTEGRATION
  200 PP(1)=-.5773502691896260
      QQ(1)=  PP(1)
      PP(2)= -PP(1)
      QQ(2)=  QQ(1)
      PP(3)=  PP(1)
      QQ(3)= -QQ(1)
      PP(4)=  PP(2)
      QQ(4)=  QQ(2)
      DO 250 I=1,4
      W(I)= 1.D0
  250 CONTINUE
      NGP = 4
      RETURN
C        4 BY 4  GAUSSIAN  INTEGRATION
  400 DUM(1) = .6521451548625460
      DUM(2) = .3478548451374540
      PP(1)  =-.8611363115940530
      QQ(1)  = PP(1)
      W(1)   = DUM(2) * DUM(2)
      PP(2)  =-.3399810435848560
      QQ(2)  = QQ(1)
      W(2)   = DUM(1) * DUM(2)
      PP(3)  =-PP(2)
      QQ(3)  = QQ(1)
      W(3)   = W(2)
      PP(4)  =-PP(1)
      QQ(4)  = QQ(1)
      W(4)   = W(1)
      PP(5)  = PP(1)
      QQ(5)  = PP(2)
      W(5)   = W(2)
      PP(6)  = PP(2)
      QQ(6)  = QQ(5)
      W(6)   = DUM(1) * DUM(1)
      PP(7)  = PP(3)
      QQ(7)  = QQ(5)
      W(7)   = W(6)
      PP(8)  = PP(4)
      QQ(8)  = QQ(5)
      W(8)   = W(2)
      PP(9)  = PP(1)
      QQ(9)  =-QQ(5)
      W(9)   = W(2)
```

```
PP(10) = PP(2)
QQ(10) = QQ(9)
W(10)  = W(6)
PP(11) = PP(3)
QQ(11) = QQ(9)
W(11)  = W(6)
PP(12) = PP(4)
QQ(12) = QQ(9)
W(12)  = W(2)
PP(13) = PP(1)
QQ(13) =-QQ(1)
W(13)  = W(1)
PP(14) = PP(2)
QQ(14) = QQ(13)
W(14)  = W(2)
PP(15) = PP(3)
QQ(15) = QQ(13)
W(15)  = W(2)
PP(16) = PP(4)
QQ(16) = QQ(13)
W(16)  = W(1)
NGP    =  16
RETURN
END
```

```
      SUBROUTINE  SSHAPE( INT )
C        CALCULATES  N(I)  AND  LOCAL  DERIVATIVES  DN(I)/DP , DN(I)/DQ
C        FOR  EIGHT  NODED  ISOPARAMETRIC  QUADRILATERAL  RING  ELEMENT
      DOUBLE PRECISION  P,Q,PP,QQ,W,N,DN
      DOUBLE PRECISION PQ,QQ
      DOUBLE PRECISION DUM
      DOUBLE PRECISION TBAR
      COMMON /SSGWRK/ P(8),Q(8),PP(16),QQ(16),W(16),N(8),DN(16),TBAR,
     1    DUM(61)
C        PP  AND  QQ  CONTAIN  COORDINATES  OF  GAUSS  INTEGRATION  POINTS
C        INT IS  THE  INDEX  OF  CURRENT  GAUSS  POINT  BEING  CONSIDERED
C        ODD  INDICES    /    CORNER NODES
      DO 10 I=1,7,2
      PQ = PP(INT) * P(I)
      QQ = QQ(INT) * Q(I)
C        CALCULATE  SHAPE  FUNCTIONS  N
      N(I) = .25D0*(1.D0+PQ)*(1.D0+QQ)*(PQ+QQ-1.D0)
C        CALCULATE  DERIVATIVE  WRT  LOCAL  COORDINATE  P  -  DN(I)/DP
      DN(I) =.25D0*P(I)*(1.D0+QQ)*(2.D0*PQ + QQ)
C        CALCULATE  DERIVATIVE  WRT  LOCAL  COORDINATE  Q  -  DN(I)/DQ
      DN(I+8)=.25D0*Q(I)*(1.D0+PQ)*(PQ+2.D0*QQ)
   10 CONTINUE
C        EVEN  INDICES    /    MIDSIDE NODES
      DO 20 I=2,8,2
      PQ = PP(INT) * P(I)
      QQ = QQ(INT) * Q(I)
      N(I) = .5D0 * ( 1.D0-PP(INT)*PP(INT) ) * ( 1.D0 + QQ ) *Q(I)*Q(I)
     1 + .5D0 * ( 1.D0-QQ(INT)*QQ(INT) ) * ( 1.D0 + PQ ) *P(I) * P(I)
      DN(I) = -PP(INT) * ( 1.D0 + QQ ) * Q(I) * Q(I)
     1    + .5D0 * ( 1.D0 - QQ(INT)*QQ(INT) ) * P(I)
      DN(I+8) = -QQ(INT) * ( 1.D0 + PQ ) * P(I) * P(I)
     1    +.5D0 * ( 1.D0 - PP(INT)*PP(INT) ) * Q(I)
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  SDUM31
C  PERFORMS PHASE I  STRESS CALCULATIONS FOR CQDRNG8 ELEMENT
      DOUBLE PRECISION  RZ,DE,G,DBAR,SIG,E,NU,RHO,ALPHA,TSUBO,GSUBE
      REAL NUS
      LOGICAL HEAT
      DIMENSION IEST(1)
      DIMENSION IECPT(1)
      COMMON /SDR2EL/ HEAT
      COMMON /MATIN/  MATID,MATFLG,ELTEMP,STRESS,SINTH,COSTH
      COMMON /HMTOUT/ FK
      COMMON /SDR2X5/ ECPT(100),EST(44)  ,RZ(16),E,G,NU,RHO,ALPHA,TSUBO,
     1  GSUBE,SIG(3),DE,DBAR
      COMMON /GPTA1/ NELEM,LAST,INCR,IE(1)
      COMMON /MATOUT/ ES,GS,NUS,RHOS,ALPHAS,TSUBOS,GSUBES,SIGS(3)
      EQUIVALENCE  ( ECPT(1),IECPT(1) ) , ( EST(1),IEST(1) )
      PI= 3.14159265358979D0
C        FILL  MATIN  COMMON  BLOCK
      MATID = IECPT(10)
      MATFLG = 1
      ELTEMP = ECPT(44)
C        CHECK FOR HEAT FORMULATION
      IF(HEAT)GO TO 9
C     I  RETRIEVE STRUCTURAL MATERIAL PROPERTIES
      CALL MAT(IECPT(1) )
C        STORE  PROPERTIES IN DOUBLE PRECISION LOCATIONS
      E = ES
      NU= NUS
      SIG(1) = SIGS(1)
      SIG(2) = SIGS(2)
      SIG(3) = SIGS(3)
      RHO = RHOS
      ALPHA = ALPHAS
      TSUBO = TSUBOS
      GSUBE = GSUBES
C        CHECK  FOR  ILLEGAL  PROPERTIES
      IF( (E.EQ.0.D0).OR.(NU.EQ.0.D0) ) GO TO 31
C        CHECK FOR POSSIBLE DIVIDE BY ZERO
      IF( NU.GE.0.5D0 ) GO TO 30
      DE = NU / (1.D0 - NU )
      G  = (1.D0 - 2.D0 * NU )/(2.D0 * (1.D0 - NU ) )
      DBAR= E * (1.D0 - NU)/( (1.D0 + NU) * (1.D0 - 2.D0 * NU ) )
      GO TO 35
C        RETRIEVE  HEAT  PROPERTIES
C    REPLACE HMAT1 CALL WITH HMAT OR PUT HMAT1 IN ROOT SEGMENT
C    9 CALL HMAT1( IECPT(1)  )
    9 CONTINUE
      E= FK
      IF( E.EQ.0.D0 ) GO TO 30
      GO TO 35
C        NU IS GE .5
   30 CALL MESAGE(-30,16,IECPT(1) )
      CALL MESAGE(-30,40,IECPT(1))
```

```
      NOGO = 1
      GO TO 35
C        E = 0. OR  NU = 0.
   31 CALL MESAGE(-30,126,IECPT(1))
      NOGO = 1
      GO TO 35
C        CHECK  NOGO  FLAG
   35 IF(NOGO.EQ.1) GO TO 5000
C        WRITE  R AND Z  COORINATES  INTO  DOUBLE  PRECISION  ARRAY  RZ
      DO 40  I=1,8
      JJ = 1 + 2*(I-1)
      J  = 13 + 4*(I-1)
      RZ(JJ  ) = ECPT(J)
      RZ(JJ+1) = ECPT(J+2)
   40 CONTINUE
C  COPY ECPT TO EST FOR PHASE II
      DO 100 J=1,11
      IEST(J) = IECPT(J)
  100 CONTINUE
      DO 110 J=12,40,4
      IEST(J) = IECPT(J)
      EST(J+1)= ECPT(J+1)
      EST(J+2)= ECPT(J+2)
      EST(J+3)= ECPT(J+3)
  110 CONTINUE
      EST(44) = ECPT(44)
C  SET FORCE AND STRESS OUTPUT WORDS (5 LOCATIONS OF 13 EACH)
      IE( 54*INCR + 18 ) = 65
      IE( 54*INCR + 19 ) = 65
 5000 RETURN
      END
```

```
      SUBROUTINE SCUM32
C   PERFORMS PHASE II STRESS CALCULATIONS FOR CQDRNG8 ELEMENT
      DOUBLE PRECISION SIG(20),SIGP(7)
      DOUBLE PRECISION P,Q,PP,QQ,W,N,DN,DNDRZ,JAC,RAD,DETERM,DUM,
     1  UGV,DSIG,INDEX,TBAR,ER,EZ,ETH,ERZ
      DOUBLE PRECISION  RZ,E,G,NU,RHO,ALPHA,TSUBO,GSUBE,SMAR,DE
      DOUBLE PRECISION DBAR
      DOUBLE PRECISION V(3),TA(9)
      REAL STR(6),PSTR(3),DIR(9)
      REAL X(3)
      INTEGER NP(5)
      INTEGER IWORD(1)
      DIMENSION IECPT(1)
      DIMENSION IS(1),IFF(1)
      COMMON /SDR2X8/ P(8),Q(8),PP(17),QQ(17),W(17),N(8),DN(16),TBAR,
     1  DNDRZ(16),JAC(2,2),INDEX(2,3),UGV(24),
     2  ER,EZ,ETH,ERZ,DSIG,DUM(4)
      COMMON /SDR2DE/ IDUM2(96),T(9)
      COMMON /SDR2X4/ WORD(39)
      COMMON /SDR2X7/ ECPT(44),RZ(16),E,G,NU,RHO,ALPHA,TSUBO,GSUBE,
     1  SMAR(3),DE,DBAR,S(100),F(100)
      EQUIVALENCE (  S(1),IS(1) ),( F(1),IFF(1) )
      EQUIVALENCE (  IECPT(1),ECPT(1)  )
      EQUIVALENCE ( WORD(1),IWORD(1) )
C   RETRIEVE DISP VECTOR AND CONVERT TO BASIC COORD SYSTEM
      DO 5 J=1,8
      CALL SDR2VR( IECPT(J+1),2,3,X,V )
      CALL TRANSD( ECPT(8+4*J),TA )
      UGV(3*J-2 ) = TA(1)*V(1) + TA(2)*V(2) + TA(3)*V(3)
      UGV(3*J-1 ) = TA(4)*V(1) + TA(5)*V(2) + TA(6)*V(3)
      UGV(3*J   ) = TA(7)*V(1) + TA(8)*V(2) + TA(9)*V(3)
    5 CONTINUE
C    PUT TEMPERATURES INTO VECTOR FOR LINTMP
C         INITIALIZE  VARIABLES  FOR  GAUSSIAN  INTEGRATION
      NGP = IECPT(11)
      CALL SD1INT(NGP,NP)
C         TOP  OF  INTEGRATION  LOOP
      DO 900  INTA =1,5
      INT = NP(INTA)
C         CALCULATE  SHAPE  FUNCTIONS  AND  LOCAL  DERIVATIVES
      CALL MSHAP8( INT )
C         CALCULATE  JACOBIAN      J = DN * RZ     STORE  IN  DUM
      CALL  GMMATD(DN,2,8,0,RZ,8,2,0,DUM)
C         STORE  DUM  IN  DOUBLY  DIMENSIONED  ARRAY  JAC(2,2)
      JAC(1,1) = DUM(1)
      JAC(1,2) = DUM(2)
      JAC(2,1) = DUM(3)
      JAC(2,2) = DUM(4)
C         CALCULATES  INVERSE  JACOBIAN
C         RETURNS  INVERSE ,  DETERMINANT ,  SINGULARITY
      CALL  INVERD( 2,JAC,2,DUM,0,DETERM,ISING,INDEX )
C         CHECK  FOR  SINGULARITY
```

```
      IF( ISTNC.EQ.2 ) GO TO 400
C        CHECK  FOR  ZERO  DETERMINANT
      IF( DETERM.EQ.0.DO ) GO TO 500
C        STORE  INVERSE  JACOBIAN  IN  VECTOR  DUM
      DUM(1)  =  JAC(1,1)
      DUM(2)  =  JAC(1,2)
      DUM(3)  =  JAC(2,1)
      DUM(4)  =  JAC(2,2)
C        CALCULATE  DNDRZ = INVERSE JAC * DN
      CALL GMMATD(DUM,2,2,0,DN,2,8,0,DNDRZ)
C        CALCULATE  RADIUS
      RAD = 0.DO
      DO  60  I=1,8
      J = 1 + 2*(I-1)
      RAD = RAD + N(I)*RZ(J)
   60 CONTINUE
C        CALCULATE  VOLUME  FOR  THIS  GAUSS  POINT
      VOL = RAD * W(INT) * DABS(DETERM)
      IF( RAD.NE.0.DO ) GO TO 80
C        ZERO  RADIUS
      CALL  MESAGE(-30,32,IECPT(1) )
      NOGO = 1
      RETURN
C        SINGULAR  JACOBIAN
  400 CALL MESAGE(-30,33,IECPT(1) )
      NOGO = 1
      IF( DETERM.NE.0.DO ) GO TO 5000
C        DETERMINANT  OF  JACOBIAN  =  0
  500 CALL  MESAGE(-30,37,IECPT(1) )
      NOGO = 1
      GO TO 5000
   80 CONTINUE
C  ZERO STRAINS
      ER =0.DO
      EZ =0.DO
      ETH=0.DO
      ERZ=0.DO
C     CALCULATE STRAINS FOR THIS GAUSS POINT
      DO 10 I=1,8
      ER = ER + DNDRZ(I)*UGV(2*I-2)
      ETH=ETH + N(I)*UGV(3*I-2)/RAD
      EZ = EZ + DNDRZ(I+8)*UGV(3*I)
      ERZ=ERZ + DNDRZ(I+8)*UGV(3*I-2) + DNDRZ(I)*UGV(3*I)
   10 CONTINUE
C  RELIEVE STRAIN DUE TO THERMAL EXPANSION
      IF(  IWORD(38).EQ.(-1) ) GO TO 19
      CALL LTMP(INT,T)
      TBAR = TBAR - TSUB0
      ER = ER - ALPHA*TBAR
      ETH= ETH- ALPHA*TBAR
      EZ = EZ - ALPHA*TBAR
   19 CONTINUE
```

```
C  CALCULATE STRESSES DUE TO REMAINING STRAIN
      K= 4*INTA-3
      SIG(K) =( ER + (ETH + EZ)*DE)*DBAR
      SIG(K+1)=(ETH+ (ER + EZ )*DE )*DBAR
      SIG(K+2)=(EZ + (ER + ETH)*DE )*DBAR
      SIG(K+3)= ERZ*G*DBAR
  900 CONTINUE
C         BOTTOM  OF  INTEGRATION  LOOP
C   EXTRAPOLATE STRESSES TO GRID POINTS  BEFORE CALCULATING INVARIANTS
      DO 910 JJ=5,17,4
      DO 910 LL=1,4
      KK = JJ + LL - 1
      DSIG = SIG(KK) - SIG(LL)
      SIG(KK) = SIG(LL) + DSIG/PP( NP(3) )
  910 CONTINUE
C  CALCULATE PRINCIPAL STRESSES AND MAX SHEAR
      DO 1200 J=1,5
      K=4*J - 3
      STR(1) = SIG(K)
      STR(2) = SIG(K+1)
      STR(3) = SIG(K+2)
      STR(4) = 0.0
      STR(5) = 0.0
      STR(6) = SIG(K+3)
      CALL PRNSTR( STR,PSTR,DIR )
C         WRITE INTO OUTPUT AREA
      JS = 1 + (J-1)*13
      IS( JS ) = IECPT(1)
      IS(JS+1) = J
      DO 1190 L=1,4
      JL = L + (J-1)*4
      S(JS + L + 1) = SIG(JL)
 1190 CONTINUE
      DO 1192 L=1,3
      S(JS+L+5) = PSTR(L)
 1192 CONTINUE
      STR(1) = ABS( .50*(PSTR(1)-PSTR(2) ) )
      STR(2) = ABS( .50*(PSTR(1) - PSTR(3) )    )
      STR(3) = ABS( .50*(PSTR(2)-PSTR(3) )    )
      S( JS+9 ) = AMAX1( STR(1),STR(2),STR(3) )
      S( JS+10 ) = DIR(1)
      S( JS+11 ) = DIR(2)
      S(JS+12) = SQRT( (PSTR(1)-PSTR(2))**2 + (PSTR(1)-PSTR(3))**2
     A  +(PSTR(2)-PSTR(3))**2 )
 1200 CONTINUE
 5000 RETURN
      END
```

```
      SUBROUTINE LTMP(INT,T)
C  LINEARIZES TEMPERATURE OVER CORRNER ELEMENT
C  RETURNS TEMP AT INTEGRATION POINT
      DOUBLE PRECISION  P,Q,PP,QQ,W,N,DN
      DOUBLE PRECISION  TBAR,PO,QO,TD
      REAL  T(1)
      COMMON /SOP2X8/ P(8),Q(8),PP(17),QQ(17),W(17),N(8),DN(16),TBAR
      TBAR = 0.DO
      DO 10  I=1,8,2
      PO =  PP(INT) * P(I)
      QO =  QQ(INT) * Q(I)
      TD =  T(I+1)
      TBAR = TBAR  +  (PO + QO) * TD
   10 CONTINUE
      TD  =  T(1)
      TBAR  = TD  +  TBAR/4.DO
      RETURN
      END
```

```
      SUBROUTINE MSHAP8( INT )
C        CALCULATES  N(I)  AND  LOCAL  DERIVATIVES  DN(I)/DP , DN(I)/DQ
C        FOR  EIGHT  NODED  ISOPARAMETRIC  QUADRILATERAL  RING  ELEMENT
      DOUBLE PRECISION P,Q,PP,QQ,W,N,DN,DNDRZ,JAC,INDEX,UGV,
     1    ER,EZ,ETH,ERZ,DSIG,DUM
      DOUBLE PRECISION PO,QO
      DOUBLE PRECISION TBAR
      COMMON /SDR2X8/ P(8),Q(8),PP(17),QQ(17),W(17),N(8),DN(16),TBAR,
     1  DNDRZ(16),JAC(2,2),INDEX(2,3),UGV(24),
     2  ER,EZ,ETH,ERZ,DSIG,DUM(4)
C        PP  AND  QQ  CONTAIN  COORDINATES  OF  GAUSS  INTEGRATION  POINTS
C        INT IS  THE  INDEX  OF  CURRENT  GAUSS  POINT  BEING  CONSIDERED
C        ODD  INDICES    /    CORNER NODES
      DO 10 I=1,7,2
      PO = PP(INT) * P(I)
      QO = QQ(INT) * Q(I)
C        CALCULATE  SHAPE  FUNCTIONS  N
      N(I) = .25D0*(1.D0+PO)*(1.D0+QO)*(PO+QO-1.D0)
C        CALCULATE  DERIVATIVE  WRT  LOCAL  COORDINATE  P  -  DN(I)/DP
      DN(I) =.25D0*P(I)*(1.D0+QO)*(2.D0*PO + QO)
C        CALCULATE  DERIVATIVE  WRT  LOCAL  COORDINATE  Q  -  DN(I)/DQ
      DN(I+8)=.25D0*Q(I)*(1.D0+PO)*(PO+2.D0*QO)
   10 CONTINUE
C        EVEN  INDICES    /    MIDSIDE NODES
      DO 20 I=2,8,2
      PO = PP(INT) * P(I)
      QO = QQ(INT) * Q(I)
      N(I) = .5D0 * ( 1.D0-PP(INT)*PP(INT) ) * ( 1.D0 + QO ) *Q(I)*Q(I)
     1 + .5D0 * ( 1.D0-QQ(INT)*QQ(INT) ) * ( 1.D0 + PO ) *P(I) * P(I)
      DN(I) = -PP(INT) * ( 1.D0 + QO ) * Q(I) * Q(I)
     1   + .5D0 * ( 1.D0 - QQ(INT)*QQ(INT) ) * P(I)
      DN(I+8) = -QQ(INT) * ( 1.D0 + PO ) * P(I) * P(I)
     1   +.5D0 * ( 1.D0 - PP(INT)*PP(INT) ) * Q(I)
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  SD1INT(NGP,NP)
C        INITIALIZES  GAUSSIAN  INTEGRATION  VARIABLES
C        ARRAYS  P AND Q  CONTAIN  LOCAL  GRID  COORDINATES
C        ARRAYS  PP, QQ, AND W  CONTAIN  GAUSS  POINT  COORDINATES  AND
C        WEIGHTING  FUNCTIONS
C        NP IS GAUSS POINT LOCATIONS TO BE USED FOR STRESS CALCULATIONS
C        FOR 2 BY 2  AND 4 X 4 GAUSS INTEGRATION A (0,0) GAUSS POINT IS ADD
C        CENTROID IS ALWAYS FIRST STRESS CALCULATED
      DOUBLE PRECISION P,Q,PP,QQ,W,N,DN,DNDRZ,JAC,INDEX,UGV,
     1    ER,EZ,ETH,ERZ,DSIG,DUM
      DOUBLE PRECISION TBAR
      INTEGER NP(5)
      COMMON /SDR2X8/ P(8),Q(8),PP(17),QQ(17),W(17),N(8),DN(16),TBAR,
     1  DNDRZ(16),JAC(2,2),INDEX(2,3),UGV(24),
     2  ER,EZ,ETH,ERZ,DSIG,DUM(4)
      P(1)=-1.D0
      Q(1)=-1.D0
      P(2)= 0.D0
      Q(2)=-1.D0
      P(3)=+1.D0
      Q(3)=-1.D0
      P(4)=+1.D0
      Q(4)= 0.D0
      P(5)=+1.D0
      Q(5)=+1.D0
      P(6)= 0.D0
      Q(6)= 1.D0
      P(7)=-1.D0
      Q(7)=+1.D0
      P(8)=-1.D0
      Q(8)= 0.D0
C        DETERMINE  NUMBER  OF  GAUSS  POINTS  TO  BE  USED
      GO TO ( 300,200,300,400 ) , NGP
C        3 BY 3  GAUSSIAN  INTEGRATION
  300 PP(1) =-.774596669241483D0
      QQ(1) =PP(1)
      DUM(1)=.555555555555556D0
      DUM(2)= .888888888888889D0
      W(1)  = DUM(1) * DUM(1)
      PP(2) = 0.D0
      QQ(2) = QQ(1)
      W(2)  = DUM(1) * DUM(2)
      PP(3) =- PP(1)
      QQ(3) =  QQ(1)
      W(3)  = W(1)
      PP(4) =  PP(1)
      QQ(4) = 0.D0
      W(4)  = W(2)
      PP(5) = 0.D0
      QQ(5) = 0.D0
      W(5)  = DUM(2) * DUM(2)
      PP(6) =  PP(3)
```

```
      QQ(6) =  0.D0
      W(6)  =  W(2)
      PP(7) =  PP(1)
      QQ(7) = -QQ(1)
      W(7)  =  W(1)
      PP(8) =  0.D0
      QQ(8) =  QQ(7)
      W(8)  =  W(2)
      PP(9) =  PP(3)
      QQ(9) =  QQ(7)
      W(9)  =  W(1)
      NP(1) = 5
      NP(2) = 1
      NP(3) = 3
      NP(4) = 7
      NP(5) = 9
      NGP   =  9
      RETURN
C        2 BY 2  GAUSSIAN  INTEGRATION
  200 PP(1)=-.577350269189626D0
      QQ(1)=  PP(1)
      PP(2)= -PP(1)
      QQ(2)=  QQ(1)
      PP(3)=  PP(1)
      QQ(3)= -QQ(1)
      PP(4)=  PP(2)
      QQ(4)=  QQ(3)
      DO 250 I=1,4
      W(I)= 1.D0
  250 CONTINUE
      NP(1) = 5
      NP(2) = 1
      NP(3) = 2
      NP(4) = 3
      NP(5) = 4
      PP(5) = 0.D0
      QQ(5) = 0.D0
      NGP = 4
      RETURN
C        4 BY 4  GAUSSIAN  INTEGRATION
  400 DUM(1) = .652145154862546D0
      DUM(2) = .347854845137454D0
      PP(1)  =-.861136311594053D0
      QQ(1)  = PP(1)
      W(1)   = DUM(2) * DUM(2)
      PP(2)  =-.339981043584856D0
      QQ(2)  = QQ(1)
      W(2)   = DUM(1) * DUM(2)
      PP(3)  =-PP(2)
      QQ(3)  = QQ(1)
      W(3)   = W(2)
      PP(4)  =-PP(1)
```

```
      QQ(4)   = QQ(1)
      W(4)    = W(1)
      PP(5)   = PP(1)
      QQ(5)   = PP(2)
      W(5)    = W(2)
      PP(6)   = PP(2)
      QQ(6)   = QQ(5)
      W(6)    = DUM(1) * DUM(1)
      PP(7)   = PP(3)
      QQ(7)   = QQ(5)
      W(7)    = W(6)
      PP(8)   = PP(4)
      QQ(8)   = QQ(5)
      W(8)    = W(2)
      PP(9)   = PP(1)
      QQ(9)   =-QQ(5)
      W(9)    = W(2)
      PP(10)  = PP(2)
      QQ(10)  = QQ(9)
      W(10)   = W(6)
      PP(11)  = PP(3)
      QQ(11)  = QQ(9)
      W(11)   = W(6)
      PP(12)  = PP(4)
      QQ(12)  = QQ(9)
      W(12)   = W(2)
      PP(13)  = PP(1)
      QQ(13)  =-QQ(1)
      W(13)   = W(1)
      PP(14)  = PP(2)
      QQ(14)  = QQ(13)
      W(14)   = W(2)
      PP(15)  = PP(3)
      QQ(15)  = QQ(13)
      W(15)   = W(2)
      PP(16)  = PP(4)
      QQ(16)  = QQ(13)
      W(16)   = W(1)
      NP(1) = 17
      NP(2) = 1
      NP(3) = 4
      NP(4) = 13
      NP(5) = 6
      PP(17)= 0.D0
      QQ(17)= 0.D0
      NGP     =  16
      RETURN
      END
```

```
      SUBROUTINE DDUM3( IOPT,IFIL,ITYPE,NE,NW,OUT )
C  FORMAT AND OUTPUT STRESSES
      REAL ROUT(2),OUT(1)
      INTEGER JOUT(2)
      EQUIVALENCE ( JOUT(1),ROUT(1) )
      GO TO ( 100,200,300) , IOPT
C        IOPT=1
  100 GO TO (110,105,105,105,105,105,105,105 ) , ITYPE
C        NO OUTPUT DESIRED
  105 IFIL = 0
      RETURN
C        REAL STRESSES  -  SORT1
  110 IFIL = -1
      NE = 1
      NW = 25
      RETURN
C        WRITE HEADER   -  IOPT=2
  200 GO TO ( 230,205,205,205,205,205,205,205 ) , ITYPE
C        NO OUTPUT DESIRED
  205 CONTINUE
      RETURN
  230 WRITE(IFIL,210)
  210 FORMAT(/T40,58HS T R E S S E S    I N    C Q D R N G R    E L E M
     1E N T S/T13,3HEID,T20,3HLOC,T26,4HSIGR,T41,5HSIGTH,T56,4HSIGZ,
     2 T72,5HSIGRZ,/T26,4HSIG1,T41,4HSIG2,T56,4HSIG3,
     3 T67,7HTAU MAX,T82,26HRZ PLANE DIRECTION COSINES,
     4  T113,10HOCTAHEDRAL/)
      RETURN
C        WRITE DATA LINES  -  IOPT=3
  300 DO 400 L=1,NW,13
      IF( (L+12).GT.NW ) GO TO 400
      ROUT(1) = OUT(L)
      ROUT(2) = OUT(L+1)
  304 GO TO ( 330,305,305,305,305,305,305,305 ) , ITYPE
C        NO OUTPUT DESIRED
  305 CONTINUE
      RETURN
C        REAL STRESSES  -  SORT1
  330 WRITE(IFIL,310) JOUT,( OUT(L+K),K=2,12)
  310 FORMAT(T7,I8,I5,1P4E15.6/T20,1P4E15.6,0P2F15.4,1PE15.6)
  400 CONTINUE
      RETURN
      END
```

X. APPENDIX B

This section contains new NASTRAN Bulk Data card descriptions for the CQDRNG8 element. These cards are used in conjunction with existing NASTRAN Bulk Data cards to model structures with CQDRNG8 elements.

NASTRAN Bulk Data cards consist of ten fields of eight columns each. The first field is a card identifier. Field 10 is used in conjunction with field 1 for continuations. If the same unique alphanumeric string appears in both fields 10 and 1 of separate cards, the cards are logically connected. The continuation string must start with + . Fields 2 through 9 are free format in the sense that data can be placed anywhere within the respective fields. The numbers above the card format box are the field numbers.

Bulk Data Deck

Input Data Card ADUM3 CQDRNG8 Dummy Element attributes

Description: Defines attributes of dummy element 3 (CQDRNG8)

Format

1	2	3	4	5	6	7	8	9	10
ADUM3	8	0	1	3	CQDRNG8				

Remarks:

1. A single ADUM3 card with entries as shown must be included in Bulk Data if the CQDRNG8 element is to be referenced.

Bulk Data Deck

Input Data Card CQDRNG8 Connections for CQDRNG8 element

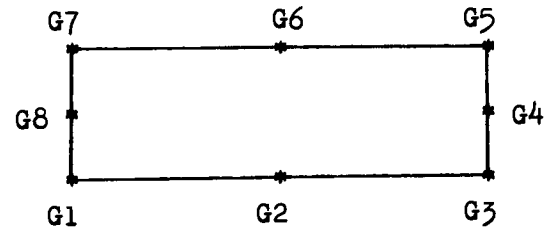
Description: Defines a linear strain axisymmetric ring element (QDRNG8) with midside grid points. This element is quadrilateral in shape and is an isoparametric formulation.

Format and Example:

1	2	3	4	5	6	7	8	9	10
CQDRNG8	EID	PID	G1	G2	G3	G4	G5	G6	+bc
CQDRNG8	369	32	111	230	141	142	143	220	+ABC

+bc	G7	G8							
+ABC	221	222							

Field	Contents
EID	Element identification number (Unique integer > 0)
PID	Identification number of a PQDRNG8 property card (Integer > 0 or blank, default is EID)
Gi	Grid point identification numbers of connection points (Integer > 0, all unique)



Remarks:

1. Element identification numbers must be unique with respect to all other EID's.
2. Connection points must be listed consecutively beginning at a corner and proceeding around the perimeter in either direction.
3. The grid points must lie in the x-z plane of the basic coordinate system, with $x=r \geq 0$.
4. This element is installed as a DUMMY element and requires one ADUM3 Bulk Data card.

Bulk Data Deck

Input Data Card PQDRNG8 CQDRNG8 property card

Description: Defines the properties of a quadratic axisymmetric ring element. Referenced by the CQDRNG8 card.

Format and Example:

1	2	3	4	5	6	7	8	9	10
PQDRNG8	PID	MID	NGP						
PQDRNG8	32	10	4						

Field	Contents
PID	Property identification number (Integer > 0)
MID	Material identification number (Integer > 0)
NGP	Order of Gaussian integration (Integer 0,2,3,4, or blank, default = 3)

Remarks:

1. All PQDRNG8 cards must have unique PID's.
2. For structural problems, a MAT1 material card is referenced. For heat transfer problems, a MAT4 material card is referenced.
3. The integration order is selected as shown:

NGP	Integration Order
2	2 by 2
3	3 by 3
4	4 by 4

The default value of 3 is recommended.